

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

Профиль подготовки: «Программное обеспечение средств вычислительной техники и автоматизированных систем»

БАКАЛАВРСКАЯ РАБОТА

на тему

**СТАРТАП-ПРОЕКТ. КРОССПЛАТФОРМЕННОЕ
МОБИЛЬНОЕ ПРИЛОЖЕНИЕ «ЗАМЕТКИ»**

Студент


(подпись, дата)

Тельнова Александра Дмитриевна
(ФИО полностью)

Руководитель


(подпись, дата)

Синев Михаил Петрович
(фамилия, инициалы)

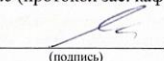
Нормоконтролёр


(подпись, дата)

Бычков Андрей Станиславович
(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 27.05.2024 г. № 11)

Заведующий кафедрой


(подпись)

М.А. Митрохин
(фамилия, инициалы)

Работа защищена с отметкой отлично (протокол заседания ГЭК от

04.06.2024 № 8)

Секретарь ГЭК


(подпись)

Е.И. Гурин
(фамилия, инициалы)

Пенза 2024

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
«19» апреля 2024 г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

1. Студент Тельнова Александра Дмитриевна гр. 20ВВП1 факультета вычислительной техники

2. Тема работы Стартап-проект. Кроссплатформенное мобильное приложение «Заметки»

Тема утверждена приказом ПГУ № 405/0 от "19" апреля 2024 г.

3. Руководитель бакалаврской работы Синев Михаил Петрович

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Разработать кроссплатформенное мобильное приложение «Заметки». Составить бизнес-модель и финансовый план. Язык программирования Dart, среда разработки Android Studio. Программа должна обладать интуитивно понятным интерфейсом, быть удобной и безопасной в использовании. Приложение должно предоставлять пользователям все необходимые функции для использования: создание, изменение, удаление и просмотр заметок, сортировка заметок, управление заметками.

5. Перечень подлежащих разработке вопросов

Требуется произвести анализ предметной области: изучить существующие информационные системы, выделить их достоинства и недостатки. Выявить потребности целевых пользователей разрабатываемого приложения. Выбрать язык программирования, среду разработки, СУБД. Определить архитектуру приложения и вспомогательные средства разработки.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ предметной области	20	19.04.2024 – 21.04.2024	
2	Обоснование актуальности разработки	16	22.04.2024 – 23.04.2024	
3	Выявление функциональных и технических требований	24	24.04.2024 – 26.04.2024	
4	Проектирование архитектуры приложения	24	27.04.2024 – 29.04.2024	
5	Выбор инструментов разработки	8	30.04.2024	
6	Разработка функционала приложения	104	01.05.2024 – 14.05.2024	
7	Тестирование и отладка приложения	20	15.05.2024 – 17.05.2024	
8	Оформление раздела по БЖД	24	18.05.2024 – 20.05.2024	
9	Оформление раздела «Стартап проект»	44	21.05.2024 – 26.05.2024	
10	Оформление пояснительной записки	40	27.05.2024 – 31.05.2024	

Дата выдачи задания " 19 " апреля 2024 г.

Руководитель бакалаврской работы  19.04.2024 Синев М.П.
(подпись, дата)

Задание к исполнению принял студент  19.04.2024 Тельнова А.Д.
(подпись, дата)

Работу к защите допустить
Декан факультета  13.06.2024 Фионова Л.Р.
(подпись, дата)

Реферат

Бакалаврская работа содержит 133 листа, 57 рисунков, 18 таблиц, 8 источников и 3 приложения

ЗАМЕТКИ, ОРГАНИЗАЦИЯ ИНФОРМАЦИИ, БАЗА ДАННЫХ, FLUTTER,
DART, КРОССПЛАТФОРМЕННОСТЬ, ANDROID STUDIO

Цель работы – разработать кроссплатформенное мобильное приложение, позволяющее пользователям фиксировать важную информацию и управлять ей.

В ходе разработки необходимо реализовать следующие функции: создание заметки, редактирование заметки, удаление заметки, просмотр заметки, назначение тегов заметке, сортировка заметок, управление заметками: перемещение заметки в корзину, восстановление заметки из корзины, перемещение заметки в архив, перемещение заметки в раздел с актуальными записями.

В результате проделанной работы было разработано кроссплатформенное мобильное приложение «Заметки», обладающее графическим интерфейсом, соответствующее поставленной задаче. Реализовано приложение, позволяющее пользователям вносить, изменять и удалять информацию, а также эффективно ей управлять. Разработанное приложение доступно для использования на смартфонах на платформах iOS и Android. Проведено тестирование и отладка каждого раздела приложения.

					ПГУ 1.090301.24.001 ПЗ		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.	Тельнова А.Д.	10.05.24	10.05.24		Стартап-проект. Кроссплатформенное мобильное приложение "Заметки" Пояснительная записка		
Провер.	Синев М.П.	10.05.24	10.05.24				
Н. Контр.	Бычков А.С.	10.05.24	10.05.24				
Утверд.	Митрохин М.А.	10.05.24	10.05.24				
					Лит.	Лист	Листов
						2	133
					ФВТ, каф. ВТ		
					гр. 208ВП1		

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

«Утверждаю»

Заведующий кафедрой

М.А. Митрохин

«19» апреля 2024 г.

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ В ФОРМЕ СТАРТАПА

Студент Тельнова Александра Дмитриевна

Направление подготовки 09.03.01 Информатика и вычислительная техника.

Профиль Программное обеспечение средств вычислительной техники и
автоматизированных систем

Форма обучения Очная

1. Тема: Стартап проект. Кроссплатформенное мобильное приложение «Заметки»,
которое будет разрабатываться данным членом команды.

Тема одобрена экспертной комиссией и утверждена приказом ректора от «19» апреля
2024 г. № 405/о.

2. Срок представления работы к защите «4» июня 2024 г.

3. Общая часть ВКР:

Подготовка компьютерной презентации бизнес-проекта (стартапа).

4. Индивидуальная часть ВКР:

4.1. Исходные данные:

Разработать кроссплатформенное мобильное приложение «Заметки». Составить бизнес-модель и финансовый план. Язык программирования Dart, среда разработки Android Studio. Программа должна обладать интуитивно понятным интерфейсом, быть удобной и безопасной в использовании. Приложение должно предоставлять пользователям все необходимые функции для использования: создание, разработка приложения, тестирование и отладка, БЖД. бизнес-план: финансовые и временные ограничения на проект (бюджет, сроки реализации, ожидаемые затраты и доходы и т.д.).

4.2. Содержание пояснительной записки ВКР (по разделам):

Анализ предметной области, анализ аналогов, постановка задачи, выбор инструментов разработки, архитектура приложения, разработка приложения, тестирование и отладка, БЖД. бизнес-план: финансовые и временные ограничения на проект (бюджет, сроки реализации, ожидаемые затраты и доходы и т.д.).

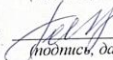
5. Перечень графического материала: UML-диаграммы (диаграмма классов, диаграмма деятельности, диаграмма вариантов использования, диаграмма последовательности)

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ предметной области	20	19.04.2024 – 21.04.2024	
2	Обоснование актуальности разработки	16	22.04.2024 – 23.04.2024	
3	Выявление функциональных и технических требований	24	24.04.2024 – 26.04.2024	
4	Проектирование архитектуры приложения	24	27.04.2024 – 29.04.2024	
5	Выбор инструментов разработки	8	30.04.2024	
6	Разработка функционала приложения	104	01.05.2024 – 14.05.2024	
7	Тестирование и отладка приложения	20	15.05.2024 – 17.05.2024	
8	Оформление раздела по БЖД	24	18.05.2024 – 20.05.2024	
9	Оформление раздела «Стартап проект»	44	21.05.2024 – 26.05.2024	
10	Оформление пояснительной записки	40	27.05.2024 – 31.05.2024	

Дата выдачи задания " 19 " апреля 2024 г.

Руководитель бакалаврской работы  19.04.2024 Синева М.П.
(подпись, дата)

Задание к исполнению принял студент  19.04.2024 Тельнова А.Д.
(подпись, дата)

Работу к защите допустить
Декан факультета  03.06.2024 Фионова Л.Р.
(подпись, дата)

Экспертное заключение
о согласовании темы ВКР в форме стартапа

Экспертная комиссия в составе декан ФВТ д.т.н., профессор Л.Р. Фионова, доцент кафедры ВТ, к.э.н., И.В. Акифьев, эксперт по разработке ООО «Ай-теко ВЦ», к.т.н. А.Н. Токарев, рассмотрев Стартап-проект. Кроссплатформенное мобильное приложение “Заметки”, Тельновой Александры Дмитриевны подтверждает его актуальность, практическую значимость и возможность коммерциализации.

Решение экспертной комиссии: выпускная квалификационная работа на тему Стартап-проект. Кроссплатформенное мобильное приложение “Заметки”, Тельновой Александры Дмитриевны может выполняться в форме стартапа (бизнес-проекта).

Председатель ЭК

д.т.н., профессор
декан ФВТ



Л.Р. Фионова

Члены ЭК

к.э.н., доцент кафедры ВТ



И.В. Акифьев

к.т.н., эксперт по разработке
ООО «Ай-теко ВЦ»



А.Н. Токарев

Содержание

Реферат	Ошибка! Закладка не определена.
Перечень сокращений и обозначений	5
Введение	6
1 Стартап-проект. Кроссплатформенное мобильное приложение «Заметки»	7
1.1 Анализ предметной области	7
1.2 Анализ аналогов	8
1.2.1 Microsoft OneNote	9
1.2.2 Notex	11
1.2.3 Evernote	13
1.3 Постановка задачи	16
1.4 Выбор среды разработки	18
1.4.1 Visual Studio Code	18
1.4.2 IntelliJ IDEA	18
1.4.3 Android Studio	19
1.5 Выбор языка разработки	20
1.6 Выбор СУБД	22
1.6.1 ObjectBox	23
1.6.2 Hive	24
1.6.3 SQLite	24
1.7 Архитектура приложения	25
1.8 Разработка приложения	29
1.8.1 Разработка модели данных	29
1.8.2 Разработка базы данных	30
1.8.3 Разработка экрана просмотра актуальных заметок	37
1.8.4 Разработка экрана просмотра заметок в архиве	38
1.8.5 Разработка экрана просмотра заметок в корзине	39
1.8.6 Разработка экрана для создания заметки	40
1.8.7 Разработка экрана для просмотра заметки	40
1.8.8 Разработка экрана для редактирования заметки	40

1.8.9	Разработка нижнего меню	41
1.8.10	Главный файл программы	41
1.9	Тестирование и отладка приложения	43
1.9.1	Тест-кейс «Создание новой заметки»	44
1.9.2	Тест-кейс «Выход из процесса создания новой заметки»	48
1.9.3	Тест-кейс «Создание новой заметки, если не все поля заполнены».....	50
1.9.4	Тест-кейс «Редактирование ранее созданной заметки»	51
1.9.5	Тест-кейс «Выход из процесса редактирования заметки».....	54
1.9.6	Тест-кейс «Просмотр ранее созданной заметки»	56
1.9.7	Тест-кейс «Удаление заметки»	58
1.9.8	Тест-кейс «Просмотр всех заметок в разделе «Актуальное»	60
1.9.9	Тест-кейс «Просмотр всех заметок в разделе «Архив»	62
1.9.10	Тест-кейс «Просмотр всех заметок в разделе «Корзина»	64
1.9.11	Тест-кейс «Перемещение заметки в актуальное»	66
1.9.12	Тест-кейс «Перемещение заметки в архив»	66
1.9.13	Сценарий «Перемещение заметки в корзину»	68
1.9.14	Тест-кейс «Фильтрация заметок по тегам»	69
2	Безопасность жизнедеятельности	72
2.1	Требования к рабочему месту	72
2.2	Освещение рабочего места	74
2.3	Микроклимат рабочего помещения.....	75
2.4	Правила при использовании мобильного устройства.....	76
2.5	Электробезопасность при работе с ПК.....	78
3	Стартап проект	80
3.1	Сегменты ЦА.....	81
3.2	Ключевые процессы	82
3.3	Каналы взаимодействия	84
3.4	Источники дохода.....	85
3.5	Ключевые ресурсы.....	86
3.6	Риски проекта.....	88

3.7 Бизнес–план.....	89
Заключение	91
Список использованных источников	92
Приложение А – UML-диаграммы.....	93
Приложение Б – Листинг программы	99
Приложение В – Презентация.....	130

Перечень сокращений и обозначений

В настоящей работе применяют следующие сокращения и обозначения
ЖК-монитор – жидкокристаллический дисплей, экран на основе жидких кристаллов.

ЗОЖ – здоровый образ жизни

ОС – операционная система

ПК – персональный компьютер

ПО – программное обеспечение

ПЭВМ – персональная электронно-вычислительная машина

СУБД – система управления базами данных

ЦА – целевая аудитория

ЭВМ – электронно-вычислительная машина

ACID (Atomicity, Consistency, Isolation, Durability) – Атомарность, Согласованность, Изоляция, Долговечность

API (Application Programming Interface) – программный интерфейс приложения

APK (Android Package Kit) – формат файла пакета для операционной системы Android

IDE (Integrated Development Environment) – интегрированная среда разработки

MVP (Minimum Viable Product) – минимально жизнеспособный продукт

MVVM (Model-View-ViewModel) – шаблон проектирования архитектуры приложения (Модель-Представление-Модель представления)

SDK (Software Development Kit) – набор инструментов для разработки программного обеспечения, объединённый в одном пакете.

SMM (Social Media Marketing) – маркетинг в социальных сетях

UML (Unified Modeling Language) – унифицированный язык моделирования

Введение

В современном мире мобильные устройства стали неотъемлемой частью нашей повседневной жизни. Смартфоны позволяют получать необходимую информацию, быть в курсе последних новостей, поддерживать общение из любой точки мира. В связи с этим появилась потребность эффективно управлять информацией, которая поступает из различных источников.

Одним из наиболее популярных инструментов для организации информации является приложение «Заметки». Оно позволяет пользователям создавать, редактировать, просматривать заметки и управлять ими с помощью мобильного устройства, которое всегда находится под рукой.

Мобильное приложение «Заметки» позволит пользователям более эффективно управлять своими задачами, улучшить свою продуктивность и сосредоточиться на важных аспектах жизни и работы.

Целью данной дипломной работы является разработка собственного мобильного кроссплатформенного приложения «Заметки», решающего проблему забывчивости, с которой сталкиваются многие люди в повседневной жизни.

В рамках работы будет проведен анализ предметной области, выявлена целевая аудитория. Так же будет проведена оценка существующих решений, их функциональных возможностей, достоинств и недостатков. В ходе работы будут определены и сформулированы требования к создаваемому приложению. Далее будет разработана архитектура приложения и интерфейс, реализованы функциональные возможности и проведено тестирование разработанного приложения.

Результатом данной работы будет создание функционального и удобного в использовании приложения «Заметки», которое поможет пользователям фиксировать важную информацию и будет являться эффективным инструментом для организации распорядка дня, четкого распределения времени.

					ПГУ1.090301.24.001 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

1 Стартап-проект. Кроссплатформенное мобильное приложение «Заметки»

1.1 Анализ предметной области

В настоящее время люди сталкиваются с большим потоком информации и высокой загруженностью. В перенасыщенном информацией мире тяжело держать в голове большие объемы важных сведений, список дел и проблем, сохранять фокус на одной задаче. При этом значительно снижается концентрация внимания и производительность.

В этом контексте приложения с возможностью фиксирования важной информации, списка дел и мыслей являются эффективным инструментом для организации распорядка дня, четкого распределения времени. Они позволяют зафиксировать все важные аспекты рабочей или личной деятельности, повысить работоспособность, свести к минимуму возможность отвлечься на посторонние раздражители, расставлять приоритеты и отслеживать прогресс выполнения задач. Электронные записи можно вести в любое время и в любом месте, так как смартфоны стали неотъемлемой частью жизни человека.

Приложения для фиксирования информации на платформах iOS и Android доступны в различных вариантах, включая бесплатные и платные, легкие и полнофункциональные. Они могут быть полезны как в личных, так и в профессиональных целях, так как справляются со сложной и многоуровневой организацией работы.

Приложение «Заметки» подходит для личного пользования. Целевая аудитория – люди, которые испытывают потребность в фиксации и организации информации. Приложение также может быть полезно людям, которым необходимо записывать свои мысли, планы на день, итоги личных или деловых встреч. Мобильное приложение позволит пользователям иметь доступ к важной информации в любом месте и в любое время, позволит обновлять и изменять ее.

					ПГУ1.090301.24.001 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

При разработке кроссплатформенного приложения «Заметки» важно оценить его функциональность, удобство использования, доступность, сохранность данных и дизайн.

Ключевые аспекты, которые необходимо рассмотреть при разработке приложения:

- функциональность. Приложение должно предоставлять все необходимые функции для фиксирования информации, такие как создание, изменение, удаление и просмотр заметок, назначение тегов и сортировка заметок. Также приложение должно предоставлять возможность перемещать заметки в архив, корзину или в раздел с актуальными записями;

- удобство использования. Приложение должно быть интуитивно понятно и удобно в использовании, чтобы максимально упростить процесс фиксирования информации и не требовать дополнительного времени на освоение функционала;

- доступность. Приложение должно предоставлять пользователям возможность установить его на свой смартфон в независимости от операционной системы мобильного устройства;

- сохранность данных. Приложение должно обеспечивать сохранность внесенных данных и отображение их пользователю;

- дизайн. Приложение должно обладать привлекательным, современным и функциональным дизайном. На этапе проектирования необходимо разработать несколько цветовых схем и дизайн-концепций.

1.2 Анализ аналогов

На сегодняшний день существует большое количество мобильных приложений, которые предоставляют широкий спектр функций для фиксирования событий, важной информации, списка дел и т. п.

					ПГУ1.090301.24.001 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

1.2.1 Microsoft OneNote

Одним из наиболее популярных приложений для фиксирования важных данных является приложение Microsoft OneNote от компании Microsoft. Программа входит в состав пакета Microsoft Office и поддерживает наличие мобильных приложений для удобства использования на смартфонах.

Приложение OneNote подходит для создания быстрых заметок и организации личной информации. OneNote представляет собой блокнот с иерархической организацией записей, как аналог обычного канцелярского блокнота.

Интерфейс приложения приведен на рисунках 1-3.

К плюсам приложения OneNote можно отнести возможность синхронизации мобильного приложения с десктопной и веб-версией приложения, возможность создавать записные книжки и разделы в записной книжке, прикреплять к записям рисунки от руки.

К минусам приложения OneNote можно отнести не самый простой функционал, нагруженный интерфейс, урезанный функционал в сравнении с десктопной версией, большой вес приложения.

Ссылка на скачивание в App Store:
<https://apps.apple.com/ru/app/microsoft-onenote/id410395246>.

Ссылка на скачивание в Google Play:
<https://play.google.com/store/apps/details?id=com.microsoft.office.onenote>.

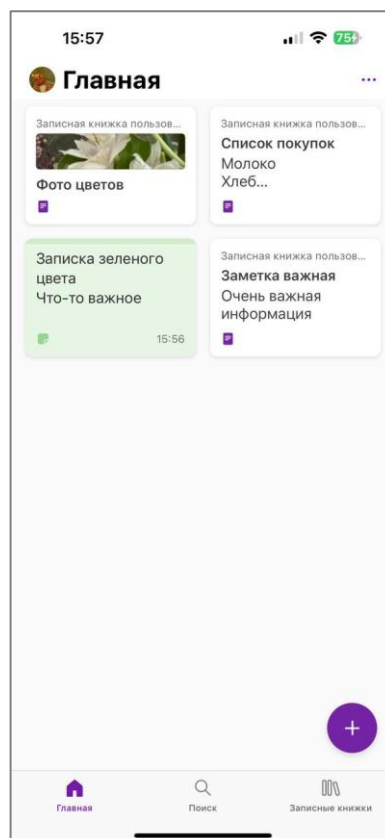


Рисунок 1 – Интерфейс приложения Microsoft OneNote

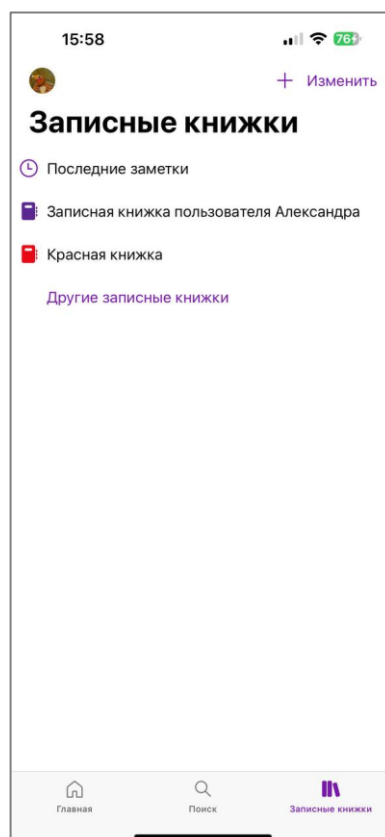


Рисунок 2 – Интерфейс приложения Microsoft OneNote

					ПГУ1.090301.24.001 ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

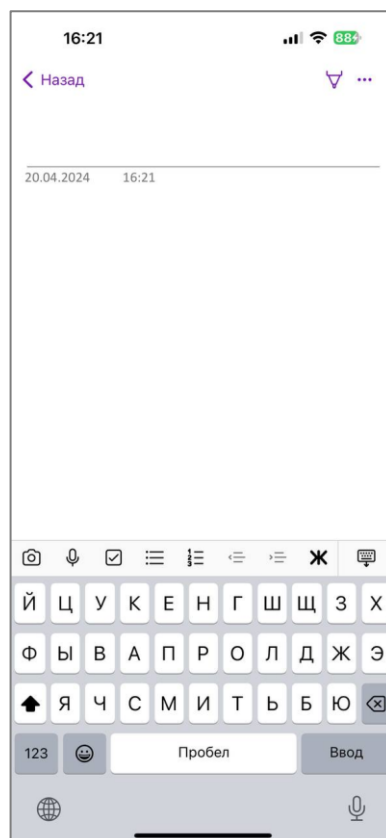


Рисунок 3 – Интерфейс приложения Microsoft OneNote

1.2.2 Notex

Другим приложением, предоставляющим функционал создания заметок, является Notex. Приложение представляет собой «доску», на которой отображаются созданные заметки в виде бумажных стикеров. Notex предлагает широкий набор функций для создания заметок и работы с ними.

Интерфейс приложения приведен на рисунках 4-5.

К плюсам приложения Notex можно отнести простой и удобный интерфейс, возможность использовать секретные ключи для шифрования заметок и фотографий, переходить между заметками свайпом влево/вправо, изменять цвет фона заметки.

К минусам приложения Notex можно отнести безвозвратное удаление заметок, неудобный выход из процесса форматирования текста, ограниченность функционала в бесплатной версии, упрощенный дизайн.

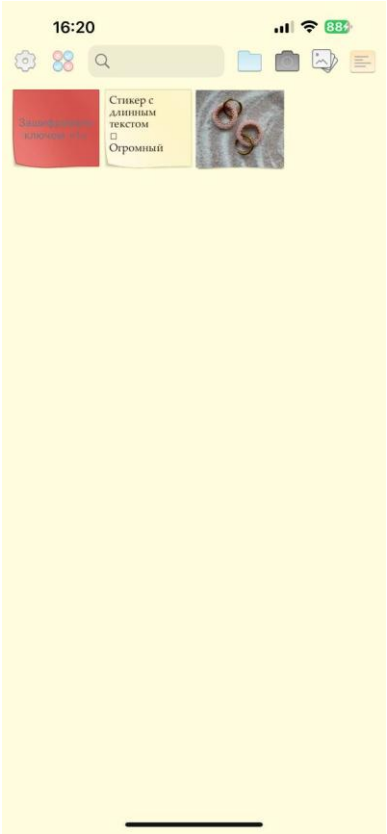


Рисунок 4 – Интерфейс приложения Notex

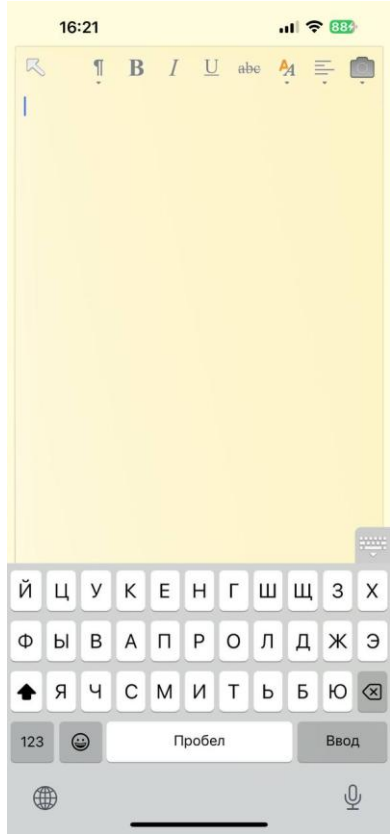


Рисунок 5 – Интерфейс приложения Notex

Ссылка на скачивание в App Store: <https://apps.apple.com/ru/app/notex-%D0%BB%D1%83%D1%87%D1%88%D0%B8%D0%B5-%D0%B7%D0%B0%D0%BC%D0%B5%D1%82%D0%BA%D0%B8/id847994217>.

1.2.3 Evernote

Evernote – популярное приложение для создания заметок, to-do листов и напоминаний. Рабочее пространство в приложении разграничено на 5 разделов: «Создать», «Заметки», «Блокноты», «Задачи», «Календарь». Приложение предоставляет широкий функционал для фиксирования текстовой и графической информации, возможность интеграции с Google Календарем.

Интерфейс приложения приведен на рисунках 6-10.

К плюсам приложения Evernote можно отнести наличие сканера документов, возможность подключения Google Календаря, синхронизацию на разных устройствах, создание блокнотов для разграничения информации, создание меток.

К минусам приложения Evernote можно отнести ограниченность бесплатной версии, перегруженность интерфейса, ограниченное количество памяти для заметок, ограниченное количество устройств для синхронизации, большой вес приложения.

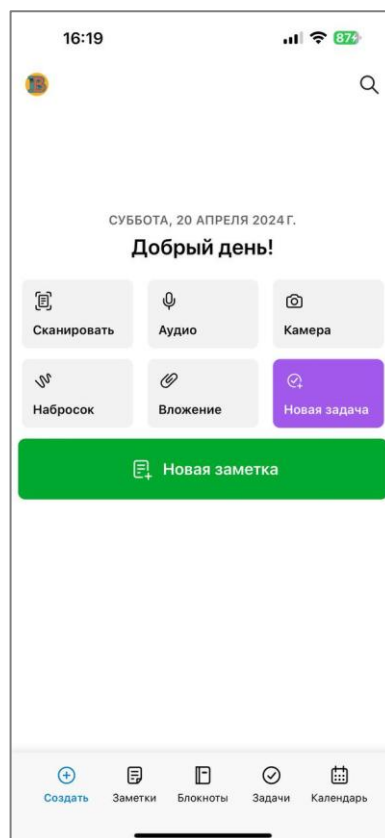


Рисунок 6 – Интерфейс приложения Evernote

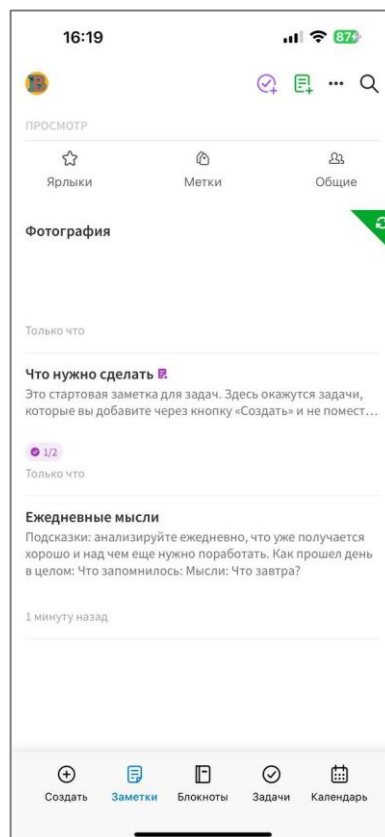


Рисунок 7 – Интерфейс приложения Evernote

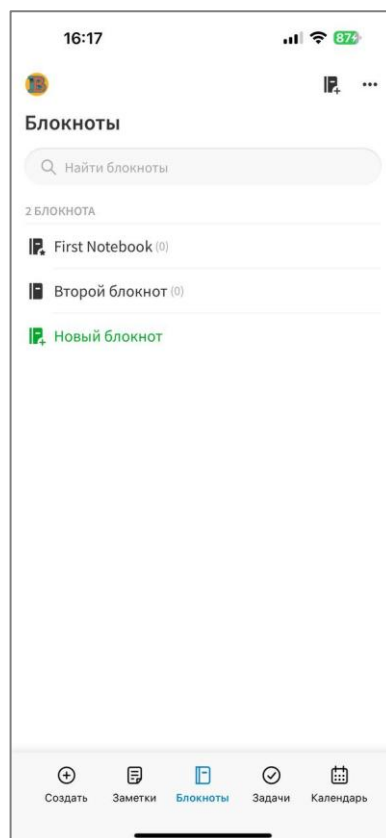


Рисунок 8 – Интерфейс приложения Evernote

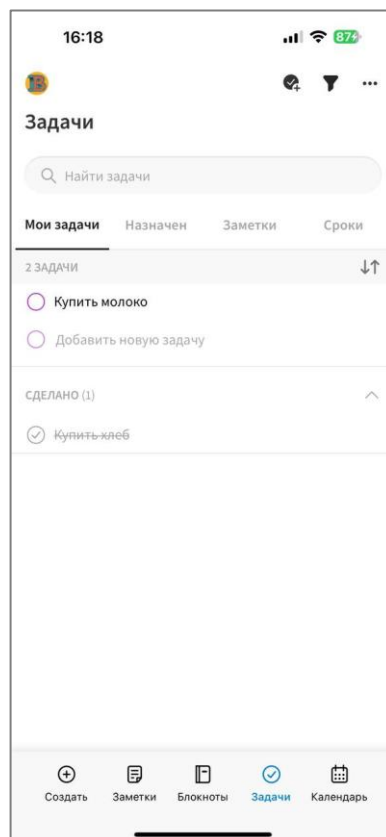


Рисунок 9 – Интерфейс приложения Evernote

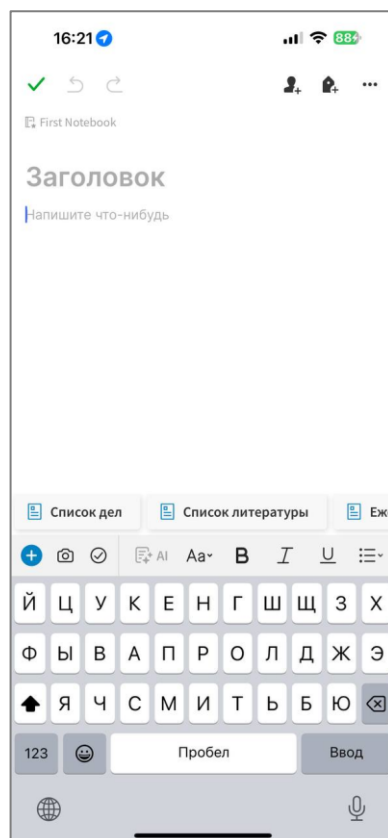


Рисунок 10 – Интерфейс приложения Evernote

Ссылка на скачивание в App Store: <https://apps.apple.com/ru/app/evernote-notes-organizer/id281796108>.

Ссылка на скачивание в Google Play: <https://play.google.com/store/apps/details?id=com.evernote>.

1.3 Постановка задачи

Задачей данного дипломного проекта является разработка кроссплатформенного мобильного приложения «Заметки».

Необходимо обеспечить пользователям возможность создания, редактирования, удаления и просмотра заметок. Предусмотреть возможность восстановления ранее удаленных заметок, назначение тегов, сортировку заметок и управление созданными заметками. Также, приложение должно обеспечивать пользователям сохранность внесенных данных.

В части функциональности приложения необходимо:

- создать механизм для добавления новой заметки;

- создать механизм для редактирования уже существующей заметки;
- создать механизм для удаления уже существующей заметки;
- реализовать возможность перемещения заметки в архив;
- реализовать возможность перемещения заметки в корзину;
- реализовать возможность перемещения заметки в раздел с актуальными заметками;
- реализовать просмотр всех ранее созданных пользователем заметок;
- реализовать просмотр конкретной заметки;
- реализовать возможность назначения тега для заметки;
- создать механизм для сортировки ранее созданных заметок по тегам.

В части разработки пользовательского интерфейса необходимо:

- разработать интуитивно понятный интерфейс;
- выбрать визуально привлекательную цветовую схему;
- обеспечить баланс между удобством и функциональностью приложения.

Технические требования к устройству

- версия iOS 13 или новее;
- версия ОС Android 11 или новее;
- наличие свободного места на устройстве в размере 30 Мб.

Тестирование и отладка:

- провести тестирование приложения на различных устройствах с разным разрешением экранов на базе ОС Android и iOS;
- обеспечить корректную работу приложения на мобильных устройствах с ОС Android и iOS;
- обеспечить совместимость и стабильность работы приложения.

1.4 Выбор среды разработки

Существует множество интегрированных сред разработки и редакторов кода для разработки мобильных приложений на iOS и Android. Некоторые из наиболее популярных сред разработки, подходящих для кроссплатформенной разработки – это Visual Studio Code, IntelliJ IDEA, Android Studio.

1.4.1 Visual Studio Code

VS Code – это текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. В отличие от Android Studio, IntelliJ IDEA и Eclipse этот редактор кода достаточно легковесный и не требует много ресурсов компьютера. VS Code включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. VS Code также поддерживает Flutter SDK.

К недостаткам VS Code можно отнести отсутствие опции создания нового Flutter-проекта. По умолчанию он может только открывать готовый. Для создания нового нужно использовать командную строку.

В качестве альтернатив VS Code можно рассмотреть IDE Android Studio и IntelliJ IDEA. Они выигрывают у VS Code по расширенному функционалу, но проигрывают по быстродействию.

1.4.2 IntelliJ IDEA

IntelliJ IDEA – интегрированная среда разработки программного обеспечения, разработанная компанией JetBrains. IntelliJ IDEA подходит для разработки программного обеспечения на многих языках программирования, в частности Java, JavaScript, Kotlin, Dart, C/C++, Python и многих других. Она также поддерживает разработку кроссплатформенных приложений и обладает множеством функций и инструментов, которые позволяют упростить процесс разработки и улучшить качество и производительность приложений.

IntelliJ IDEA обладает большим набором профессиональных

					ПГУ1.090301.24.001 ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

инструментов: средства для отладки, рефакторинга кода, готовые решения для работы с базами данных, инструменты для развёртывания и сборки приложений и многое другое. Дополнительно IDE предлагает множество вариантов исправления кода на более простые и современные методы.

К недостаткам IntelliJ IDEA относится ограниченность бесплатной версии.

У программы есть две версии:

- IntelliJ IDEA Community Edition. Бесплатная версия с ограниченной функциональностью. Она не поддерживает большинство фреймворков;

- IntelliJ IDEA Ultimate Edition. Платное расширенное издание, предоставляющее более обширный набор инструментов для профессиональной разработки.

Функциональность бесплатной версии сильно ограничена, а приобрести расширенную версию на сегодняшний день невозможно, так как компания JetBrains приостановила продажи продуктов и услуг в России.

IntelliJ IDEA потребляет много ресурсов. При работе с большими проектами среда может требовать большого количества оперативной памяти и ресурсов процессора, что может сказаться на производительности машины.

1.4.3 Android Studio

Android Studio – это интегрированная среда разработки, созданная компанией Google. Она имеет множество функций и инструментов, которые делают процесс разработки более легким, эффективным и удобным. Одной из важных функций является наличие виртуальных устройств (эмуляторов).

Основные характеристики:

Android Studio может быть свободно загружена и использоваться любым пользователем. Она доступна для Windows, macOS и Linux и поддерживает программирование на Java, C, C ++, Kotlin, XML, а также интеграцию с инструментами Google, такими как Firebase и Google Cloud.

Установив Flutter SDK, можно писать еще и на Dart, что позволяет вести кроссплатформенную разработку программного обеспечения.

Android Studio позволяет быстро создавать макеты, перетаскивая элементы пользовательского интерфейса в редактор визуального дизайна. Этот редактор может проверять макеты на различных устройствах и версиях Android. Он может динамически изменять размер макетов, чтобы они отображались в соответствии с размером экрана.

Android Studio поставляется с анализатором APK для проверки состава APK (исполняемого файла приложения). Это сокращает время отладки файлов DEX (содержащих код) и других ресурсов. Это уменьшает размер APK, что упрощает установку на устройства с низким объемом памяти.

Android Studio поставляется с эмулятором различных устройств, который помогает тестировать приложения на разных устройствах без использования реального физического устройства.

Android Studio интегрируется с системой контроля версий Git, что позволяет разработчикам легко управлять и обновлять исходный код в различных ветках.

К недостаткам Android Studio относится потребность в большом количестве оперативной памяти. Так же Android Studio может быть сложна в использовании для новичков, так как обладает сложным интерфейсом и требует некоторого опыта в программировании.

Android Studio является лучшим выбором для разработки приложений на платформе iOS и Android, благодаря большому количеству инструментов и поддержке Flutter SDK. Несмотря на высокие требования к аппаратному обеспечению и сложность интерфейса, Android Studio является удобной IDE для кроссплатформенной разработки.

1.5 Выбор языка разработки

Для разработки кроссплатформенных приложений (iOS и Android) в среде разработки Android Studio необходим Flutter SDK.

					ПГУ1.090301.24.001 ПЗ	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

Flutter – это бесплатный и открытый фреймворк, созданный и выпущенный компанией Google.

Главная особенность в работе с кроссплатформенным движком – это использование одной кодовой базы на языке Dart вместо нескольких для Android и iOS платформ. Flutter позволяет разрабатывать кроссплатформенные проекты сразу и под iOS, и под Android.

К плюсам Flutter можно отнести высокую производительность сборок, простой и лаконичный язык программирования Dart, а также быструю компиляцию и сборку проекта.

Также достоинством является большой набор UI-компонентов и библиотек, наличие обширной документации и обучающие видео от Google.

Один из явных минусов Flutter в том, что сборка приложения для iOS, запуск симулятора и запуск на реальном iPhone невозможен без XCode, который работает только на машинах с MacOS. Для решения проблемы нужно иметь XCode, плагин на Android Studio сам подцепит симулятор и реальный девайс.

Другой минус в том, что увидеть результат работы можно, только запустив приложение, так как верстка экранов создается с помощью кода.

Так как во Flutter используется Dart – современный объектно-ориентированный язык программирования, то он будет выбран в качестве языка программирования для реализации кроссплатформенного мобильного приложения «Заметки».

В основе Dart – классы с синтаксисом в стиле C, при этом программа может компилироваться как в собственный код, так и в JavaScript.

Dart является альтернативой JavaScript, которая позволила создавать более сложные, приложения, облегчая при этом процесс разработки.

К достоинствам языка относится:

– поддержка асинхронного программирования. За счёт поддержки функционального стиля Dart пригоден для асинхронного программирования, при котором операции выполняются, не блокируя основной поток работы

программы. Благодаря этому пользовательский интерфейс продолжает работать нормально, пока выполняются сложные операции;

- простой и понятный синтаксис;
- богатая стандартная библиотека. Она включает функции для работы с коллекциями, обработки ошибок, работы с сетью и многое другое;
- мощные инструменты для разработки пользовательского интерфейса. Dart поставляется с набором библиотек, которые облегчают создание интерфейсов для мобильных и веб-приложений. Эти библиотеки включают виджеты, макеты и стили.

1.6 Выбор СУБД

Системы управления базами данных (СУБД) играют важную роль при разработке программного обеспечения, так как они обеспечивают эффективное управление данными и их хранение.

Необходимо тщательно подойти к выбору СУБД, так как она является одним из ключевых компонентов при разработке приложений, обеспечивающих хранение важной информации. Этот аспект особенно важен при разработке мобильного приложения «Заметки», так как оно должно обеспечивать надежность и сохранность информации, внесенную пользователем. Выбранный программный продукт должен удовлетворять как текущим, так и будущим потребностям создаваемого проекта.

Причины, по которым СУБД важны в проекте:

- управление данными. СУБД позволяют управлять большими объемами данных, включая их создание, изменение, удаление и чтение. Они предоставляют инструменты для организации данных, такие как таблицы, индексы и ключи, что упрощает поиск и обработку информации;
- безопасность данных. СУБД обеспечивают безопасность данных, позволяя контролировать доступ к ним. Это особенно важно в проекте, где данные являются конфиденциальными;

- масштабируемость. СУБД могут масштабироваться вместе с ростом проекта, поддерживая большое количество пользователей и обрабатывая большие объемы данных;
- сохранение данных. СУБД гарантируют сохранность данных, автоматически сохраняя изменения и предотвращая потерю информации;
- долгосрочное хранение. СУБД предназначены для долгосрочного хранения данных, что позволяет использовать информацию в течение длительного времени.

Так как для разработки кроссплатформенного мобильного приложения выбран фреймворк Flutter, то наиболее популярными и используемыми локальными базами данных для Flutter являются ObjectBox, Hive, SQLite.

1.6.1 ObjectBox

Одна из самых востребованных локальных баз данных Flutter, ObjectBox - это база данных NoSQL, которая лучше всего подходит для мобильных устройств и интеллектуальных устройств Интернета вещей. Это относительно более быстрая и высокопроизводительная база данных, использующая объектно-ориентированный подход, устраняющий необходимость сопоставления объектов с таблицами.

ObjectBox использует минимум ресурсов процессора, памяти и аккумулятора, что делает приложение не только эффективным, но и устойчивым. Благодаря локальному хранению данных на устройстве ObjectBox помогает сократить расходы на облако и создать приложение, не зависящее от Интернет-подключения.

ObjectBox поставляется со встроенной синхронизацией данных для децентрализованных периферийных вычислений и интуитивно понятным языком запросов, а также поддерживает транзакции ACID (атомарность, согласованность, изоляция и долговечность).

У ObjectBox есть несколько недостатков: сложна в использовании по сравнению с другими вариантами, обладает меньшей поддержкой сложных отношений между объектами.

					ПГУ1.090301.24.001 ПЗ	Лист
						23
Изм.	Лист	№ докум.	Подпись	Дата		

1.6.2 Hive

Hive одна из наиболее широко используемых локальных баз данных Flutter. Некоторые из причин ее популярности – сверхбыстрая производительность и легкое устройство. Она написана на чистом Dart и позволяет хранить данные в HiveObject, облегчая отношения между объектами.

Эта база данных поставляется с простым, но мощным и интуитивно понятным API, который может обновлять пользовательский интерфейс в режиме реального времени.

Эта библиотека подходит для приложений с простыми и быстрыми требованиями к хранению данных. Кроме того, если приложению не требуется синхронизация данных в реальном времени, то Hive – хороший вариант для разработки приложений Flutter.

К недостаткам Hive можно отнести отсутствие поддержки сложных запросов к базе данных, а также могут отсутствовать некоторые функции, предлагаемые аналогами. Кроме этого, обновление существующей базы данных может потребовать от разработчика соблюдения определенных правил совместимости, что может занять много времени.

1.6.3 SQLite

SQLite – компактная встраиваемая СУБД. SQLite, имеет небольшой вес и предназначена для мобильных устройств. Простая в использовании, SQLite дает полный контроль над запросами и взаимосвязями приложения.

SQLite не использует парадигмы клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа. В качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу.

Несколько процессов или потоков могут одновременно читать данные из одной базы. Запись в базу можно осуществить только в том случае, если

никаких других запросов в данный момент не обслуживается, в противном случае попытка записи оканчивается неудачей.

Эта база данных поставляется с простым и реактивным API, почти со всеми стандартами SQL и поддерживает транзакции ACID.

SQLite идеально подходит для приложений как среднего, так и большого размера.

Однако сложные запросы и большие наборы данных могут вызвать некоторые проблемы с производительностью. А также SQLite может не хватать скорости по сравнению с опциями NoSQL.

Несмотря на перечисленные недостатки, SQLite является лучшим выбором для разработки кроссплатформенных приложений на Flutter. SQLite легковесна и позволяет быстро вносить и извлекать данные, проста в настройке и интеграции с приложениями. SQLite является стабильной и надежной базой данных, обеспечивающей высокую сохранность данных. Также SQLite поддерживает транзакции, что позволяет управлять параллельным доступом к базе данных.

1.7 Архитектура приложения

Для разработки кроссплатформенного мобильного приложения заметки выбран паттерн MVVM.

MVVM (Model-View-ViewModel) – это паттерн проектирования пользовательского интерфейса, состоящий из трех ключевых элементов: модель (Model), представление (View) и представление модели (ViewModel). Каждый из этих элементов имеет свои задачи в приложении.

Модель (Model)

Модель представляет собой часть приложения, которая содержит бизнес-логику и данные. Она не зависит от представления и может быть использована в различных частях приложения. Модель отвечает за управление данными и логикой приложения.

Представление (View)

					ПГУ1.090301.24.001 ПЗ	Лист
						25
Изм.	Лист	№ докум.	Подпись	Дата		

Представление – это визуальная часть приложения, которая отображает данные модели и предоставляет пользователю возможность взаимодействовать с приложением.

Представление модели (ViewModel)

ViewModel является посредником между представлением и моделью. Она содержит логику, необходимую для обработки событий, происходящих в представлении, и для обновления данных модели. ViewModel также может содержать дополнительные свойства и методы, необходимые для работы представления.

Преимущества использования MVVM:

- улучшенная тестируемость: благодаря разделению ответственности между различными компонентами приложения становится проще тестировать каждый компонент отдельно;
- поддержка переиспользования кода: использование шаблонов проектирования позволяет создавать повторно используемые компоненты, что упрощает разработку и поддержку приложений;
- лучшая поддержка UI: благодаря тому, что представление отделено от бизнес-логики, становится проще поддерживать и обновлять пользовательский интерфейс без изменения основной функциональности приложения;
- простота разработки: использование MVVM позволяет разработчикам сосредоточиться на написании бизнес-логики и логики представления, не беспокоясь о том, как эти элементы будут взаимодействовать друг с другом.

Архитектура приложения состоит из двух основных частей: клиентская часть и база данных.

Клиентская часть приложения — это та часть программного обеспечения, с которой взаимодействует пользователь. Клиент представляет собой графический интерфейс и его компоненты, которые реагируют на действия пользователя.

Клиентская часть отвечает за отображение информации, взаимодействие с пользователем через элементы управления (кнопки, поля ввода, меню и т. д.), обработку событий (например, нажатие кнопок), а также обращение к базе данных для записи или получения данных.

База данных – совокупность данных, организованных по определённым правилам и предусматривающих общие принципы описания, хранения и манипулирования данными.

Приложение будет состоять из следующих экранов:

- экран со списком всех актуальных заметок, созданных пользователем (страница «Мои заметки»);
- экран со списком заметок, находящимися в архиве (страница «Архив»);
- экран со списком заметок, находящимися в корзине (страница «Недавно удаленные»);
- экран просмотра заметки;
- экран создания заметки (страница «Добавление заметки»);
- экран редактирования заметки (страница «Редактирование заметки»).

Экран со списком актуальных заметок будет стартовым при первом и последующем запуске приложения и будет отображать все заметки, созданные пользователем, а также содержать основные элементы управления: кнопку создания новой заметки, меню с кнопками для перехода в архив и корзину, кнопку для сортировки заметок. На карточке каждой заметки будут доступны кнопки для редактирования заметки, перемещения в архив и корзину.

Экран со списком заметок, находящимся в архиве, будет отображать заметки, перенесенные пользователем на длительное хранение. На карточке заметки дополнительно будут добавлены элементы управления, позволяющие отредактировать заметку, переместить заметку из архива на главный экран с актуальными записями или переместить ее в корзину.

					ПГУ1.090301.24.001 ПЗ	Лист
						27
Изм.	Лист	№ докум.	Подпись	Дата		

Экран со списком заметок, находящимся в корзине, будет отображать заметки, удаленные пользователем. На карточке заметки будут располагаться инструменты, позволяющие окончательно удалить заметку, восстановить заметку из корзины и переместить ее на главный экран или переместить заметку в архив.

Экран просмотра заметки будет содержать заголовок заметки, основной текст заметки и кнопку «Назад».

Экран для создания заметки будет содержать область для ввода заголовка заметки, основного текста заметки, кнопку, вызывающую модальное окно со списком тегов, доступных для назначения, кнопку «Назад» и кнопку «Сохранить».

Экран для редактирования заметки будет содержать область для изменения заголовка заметки, основного текста заметки с ранее введенными данными, кнопку «Назад» и кнопку «Сохранить».

Диаграмма вариантов использования приведена на рисунке 11.

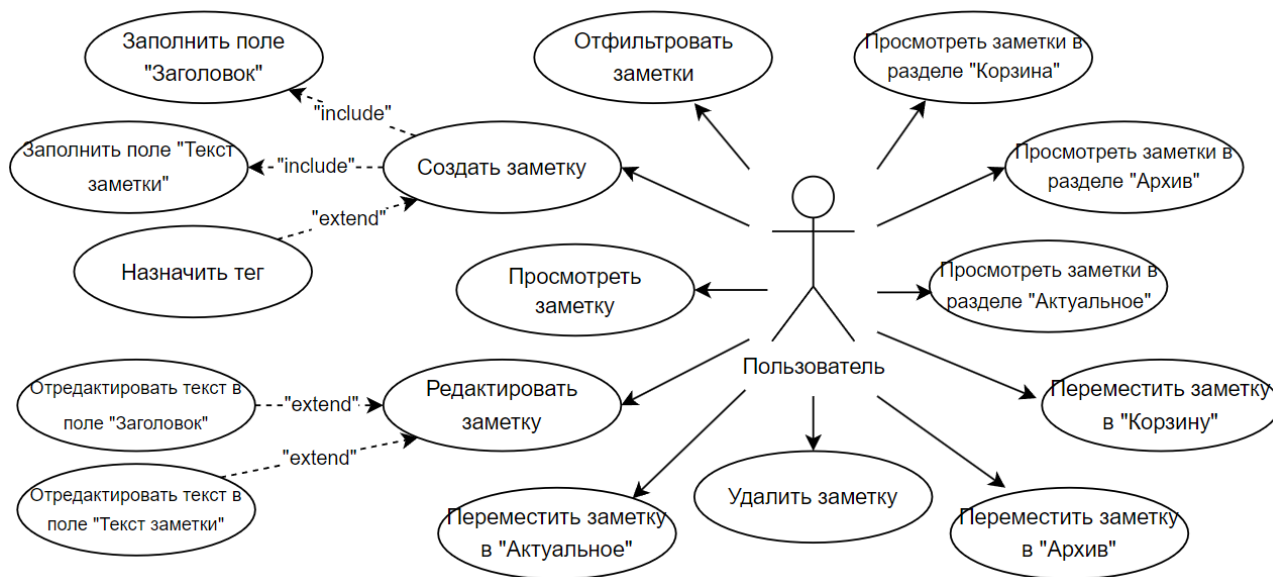


Рисунок 11 – Диаграмма вариантов использования

Для хранения всей информации необходимо разработать в базе данных таблицу, содержащую сведения о внесенных пользователями записях: идентификатор заметки, заголовок заметки, текст заметки, дата и время

создания заметки, признак нахождения заметки в архиве, признак нахождения заметки в корзине, тег заметки.

1.8 Разработка приложения

1.8.1 Разработка модели данных

Для описания модели заметки создан класс NoteModel, содержащий такие поля, как идентификатор заметки, заголовок заметки, основной текст заметки, дата создания, признак нахождения заметки в архиве, признак нахождения заметки в корзине, тег заметки.

Идентификатор заметки (id) – это уникальный номер записи в базе данных, имеет тип «целое число». Он никогда не повторяется, что позволяет однозначно взаимодействовать с записью в базе данных.

При создании новой заметки ей присваивается уникальный идентификатор, при обновлении имеющейся записи идентификатор сохраняется прежним.

Заголовок заметки (title) отображает название заметки, которое будет введено пользователем. Призван отобразить ключевую информацию/тему записи, имеет тип «строка».

Основной текст заметки (body) отображает основной текст заметки, всю важную информацию, которая будет введена пользователем, имеет тип «строка».

Дата создания заметки (date_created) отображает дату и время создания заметки пользователем, имеет тип «дата-время».

Признак нахождения заметки в архиве (is_archived) – позволяет отделить актуальную для пользователя часть записей от тех, которые перемещены в архив. Параметр имеет тип «целое число» и выставляется в «1», если заметка находится в архиве, иначе в «0».

Признак нахождения заметки в корзине (is_deleted) – позволяет отделить актуальную для пользователя часть записей от тех, которые

перемещены в корзину. Параметр имеет тип «целое число» и выставляется в «1», если заметка находится в корзине, иначе в «0».

Тег заметки (tag) описывает категорию, к которой принадлежит заметка, например, «Личное» или «Работа», имеет тип «строка». Тег отображается на карточке заметки в виде цветного бейджа и позволяет определить принадлежность заметки к какой-либо категории.

1.8.2 Разработка базы данных

Для хранения заметок пользователя с помощью функции initDB создана таблица all_notes, содержащая всю информацию о внесенных данных. Атрибуты таблицы all_notes описаны в таблице 1.

Таблица 1 – Атрибуты таблицы all_notes

Имя столбца	Описание	Тип данных	Роль атрибута	Пример
id	Идентификатор записи	INTEGER	PRIMARY KEY AUTOINCREMENT	15
title	Заголовок заметки	BLOB		Важная информация
body	Основной текст заметки	BLOB		На последней встрече с коллегами выяснилась очень важная информация
date_created	Дата создания заметки	DATE		2024-05-08 11:05:27
is_archived	Признак нахождения заметки в архиве	INTEGER		1
is_deleted	Признак нахождения заметки в корзине	INTEGER		0
tag	Тег заметки, категория	STRING		Работа

В процессе разработки приложения возникает необходимость расширять исходную таблицу. Для реализации этой возможности создана функция `_onUpgrade`. Функция позволяет добавить в исходную таблицу новый столбец и обновить версию самой таблицы.

Для занесения в базу данных новой записи создана функция `addNewNote`. Функция сохраняет в таблице `all_notes` в соответствующих полях всю информацию о заметке: идентификатор заметки, заголовок заметки, основной текст заметки, дата создания, признак нахождения заметки в архиве, признак нахождения заметки в корзине, тег заметки.

Диаграмма последовательности приведена рисунке 12.

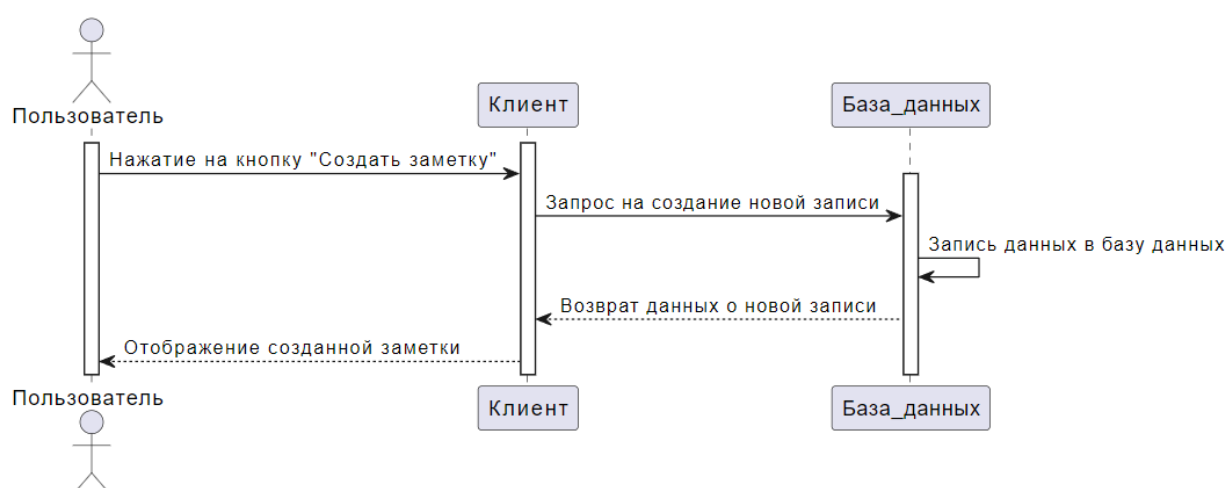


Рисунок 12 – Диаграмма последовательности «Создание новой заметки»

Функция `getNotes` позволяет отобразить в разделе «Актуальное» все заметки, которые не находятся в архиве или корзине и актуальны для пользователя на данный момент, а также реализует функцию фильтрации заметок по тегам. Функция принимает на вход три параметра: признак нахождения заметки в архиве, признак нахождения заметки в корзине, тег заметки, при этом из таблицы выбираются записи, у которых параметры `is_archived` и `is_deleted` выставлены в «0». Параметр `tag` позволяет получить записи с определенным тегом.

Диаграмма последовательности приведена рисунке 13.

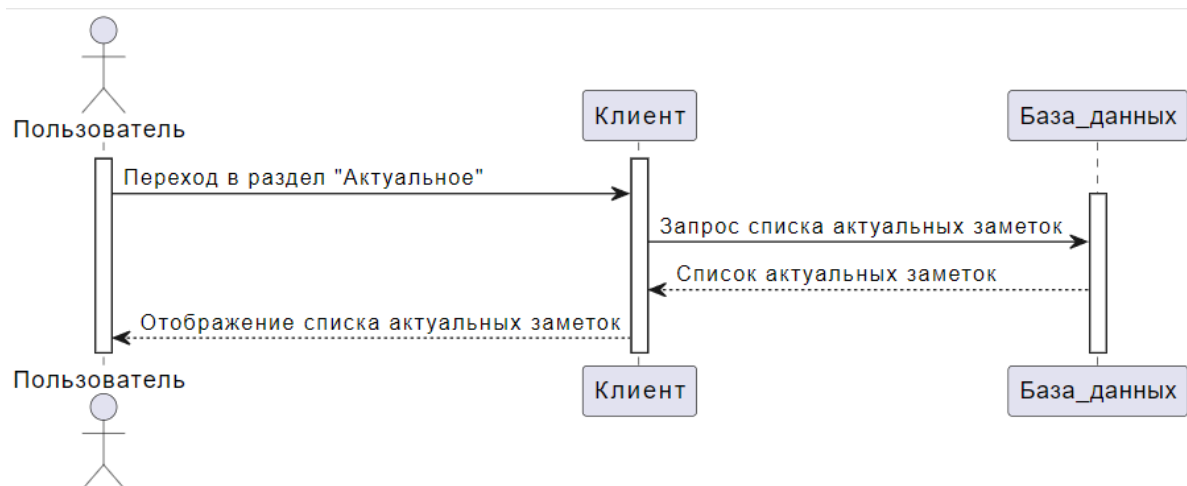


Рисунок 13 – Диаграмма последовательности «Отображение записей в разделе «Актуальное»

Для удаления из базы данных имеющейся записи создана функция deleteNote. Функция принимает на вход параметр id – идентификатор записи и удаляет из таблицы all_notes требуемую запись.

Диаграмма последовательности приведена рисунке 14.

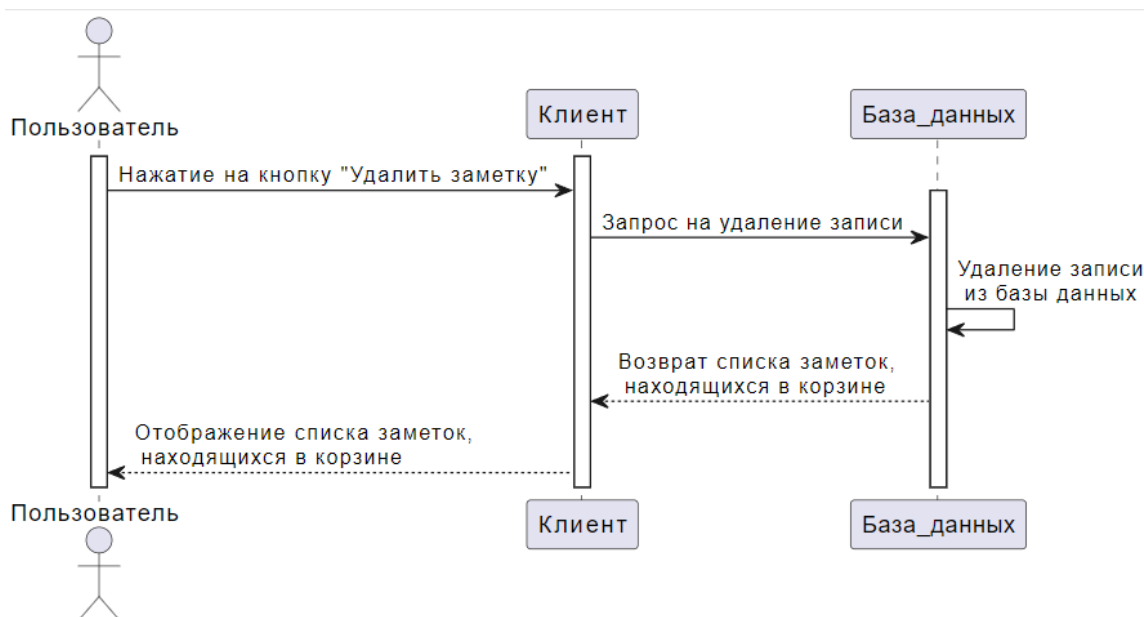


Рисунок 14 – Диаграмма последовательности «Удаление заметки»

Для обновления в базе данных имеющейся записи создана функция updateNote. Функция принимает на вход модель данных NoteModel и

обновляет все поля таблицы all_notes по идентификатору изменяемой заметки.

Диаграмма последовательности приведена рисунке 15.

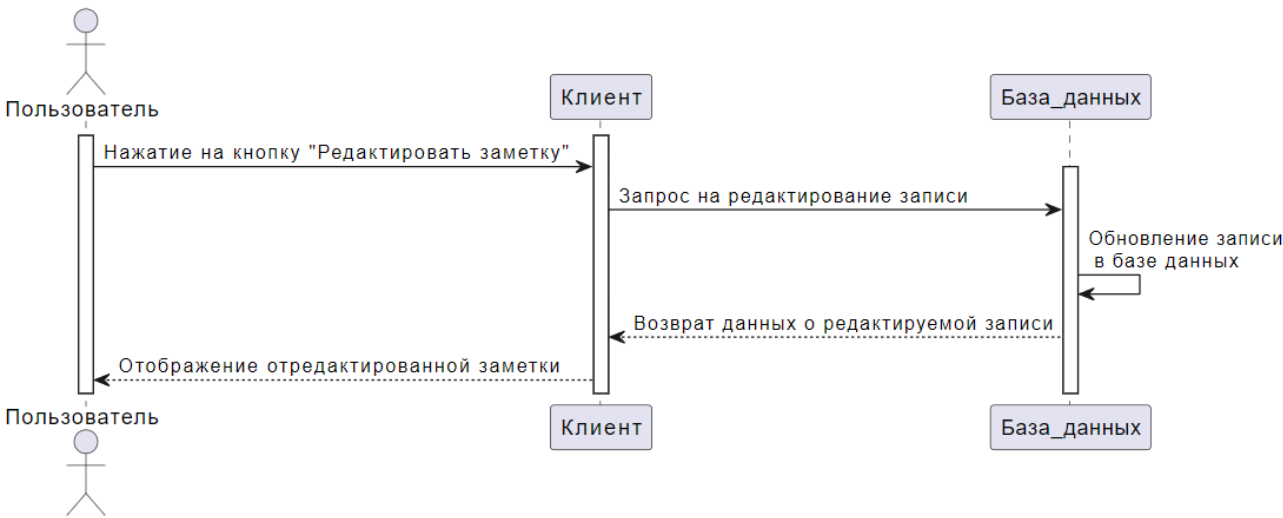


Рисунок 15 – Диаграмма последовательности «Редактирование заметки»

Функция toArchive позволяет переместить заметку в архив. Функция обновляет поле is_archived таблицы all_notes по идентификатору перемещаемой заметки.

Диаграмма последовательности приведена рисунке 16.

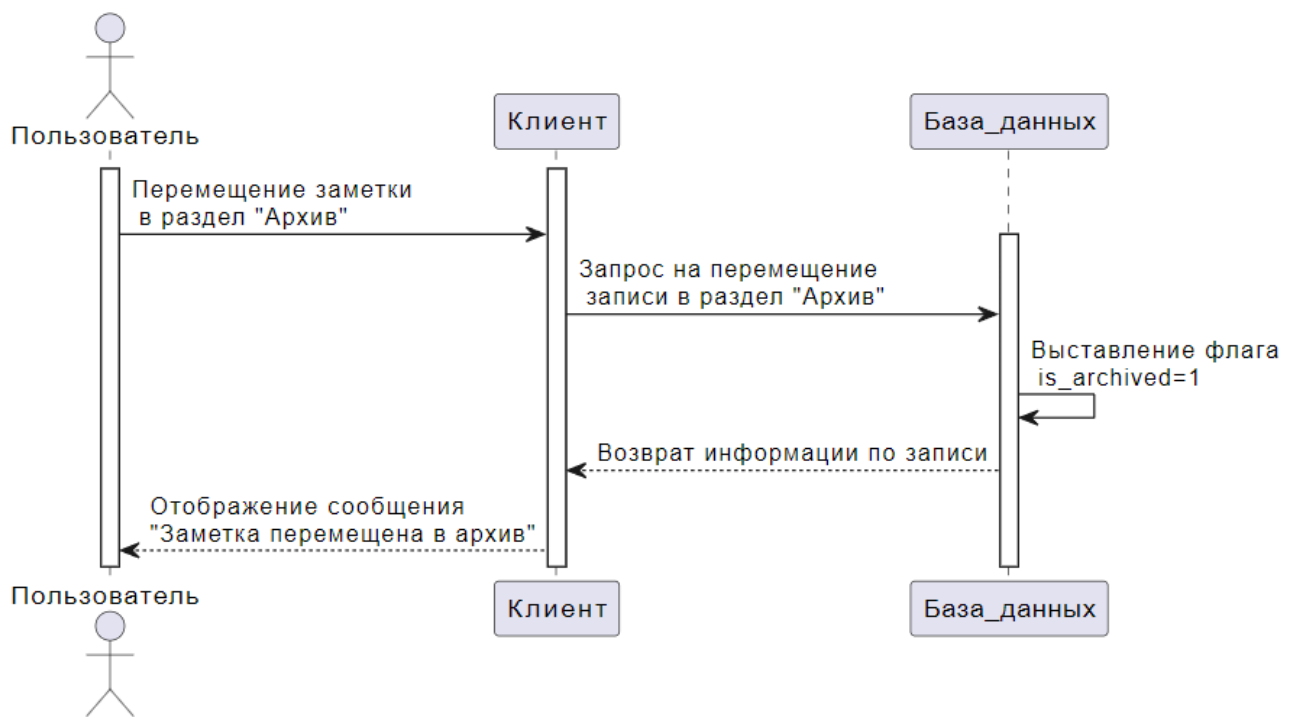


Рисунок 16 – Диаграмма последовательности «Перемещение заметки в раздел «Архив»

Функция toDelete позволяет переместить заметку в корзину. Функция обновляет поле is_deleted таблицы all_notes по идентификатору перемещаемой заметки.

Диаграмма последовательности приведена рисунке 17.

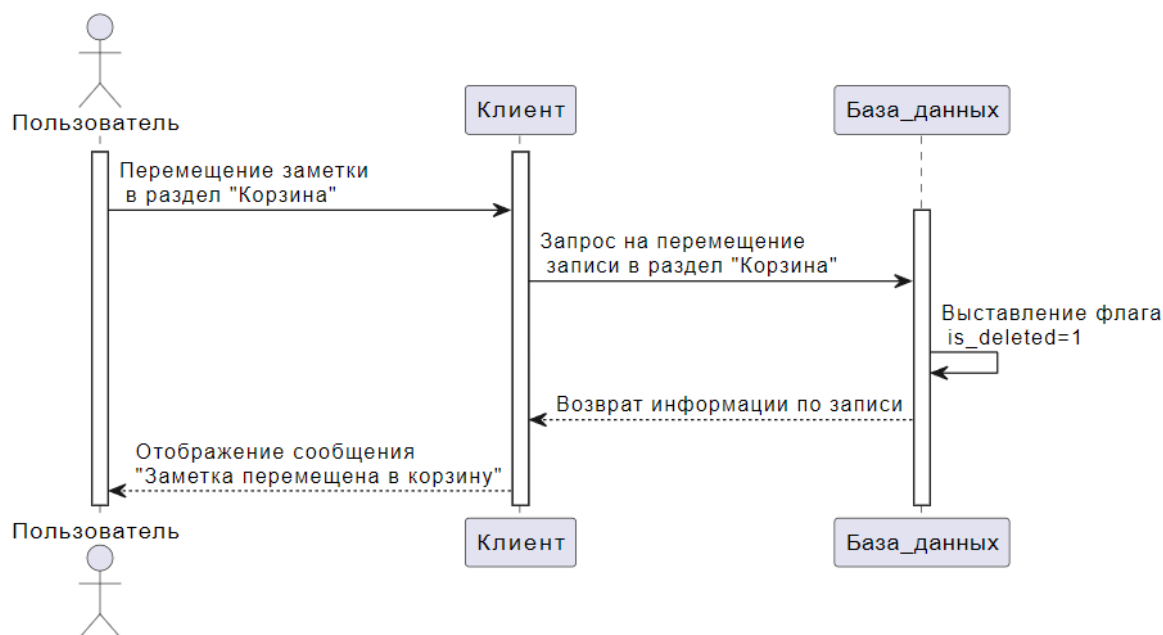


Рисунок 17 – Диаграмма последовательности «Перемещение заметки в раздел «Корзина»

Функция toActual позволяет переместить заметку в раздел «Актуальное». Функция обновляет поля is_archived и is_deleted таблицы all_notes по идентификатору перемещаемой заметки.

Диаграмма последовательности приведена рисунке 18.

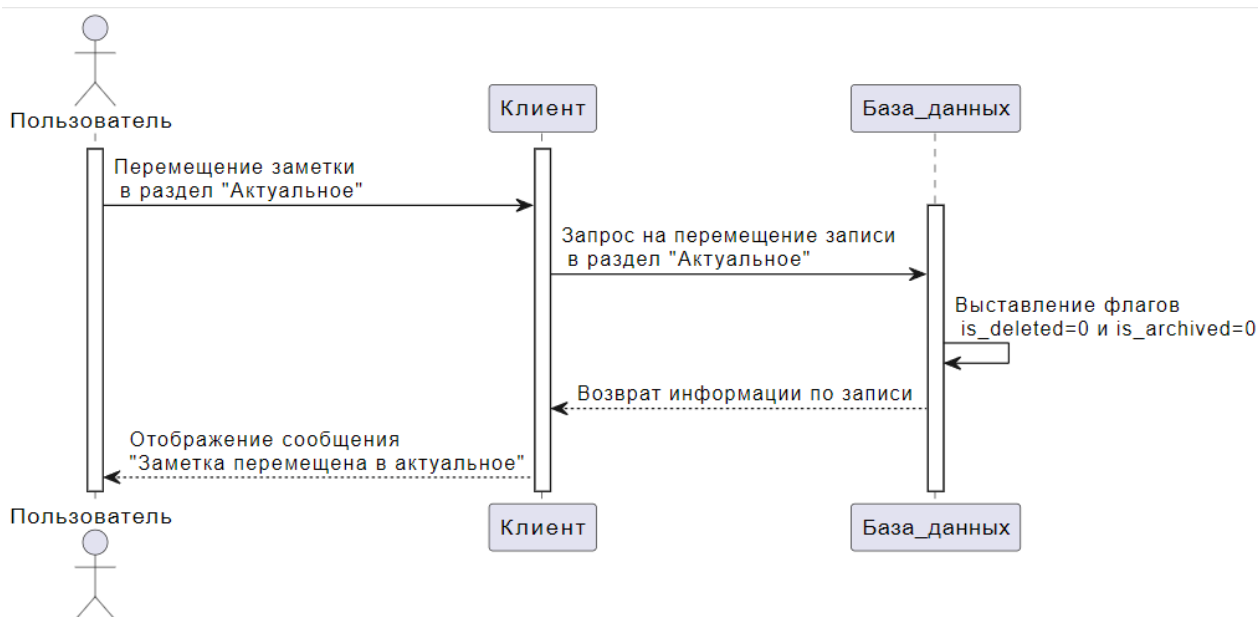


Рисунок 18 – Диаграмма последовательности «Перемещение заметки в раздел «Актуальное»

Функция `getArchived` позволяет отобразить в разделе «Архив» все заметки, которые находятся в архиве. Функция принимает на вход параметр с признаком нахождения заметки в архиве, при этом из таблицы выбираются записи, у которых этот параметр выставлен в «1».

Диаграмма последовательности приведена рисунке 19.

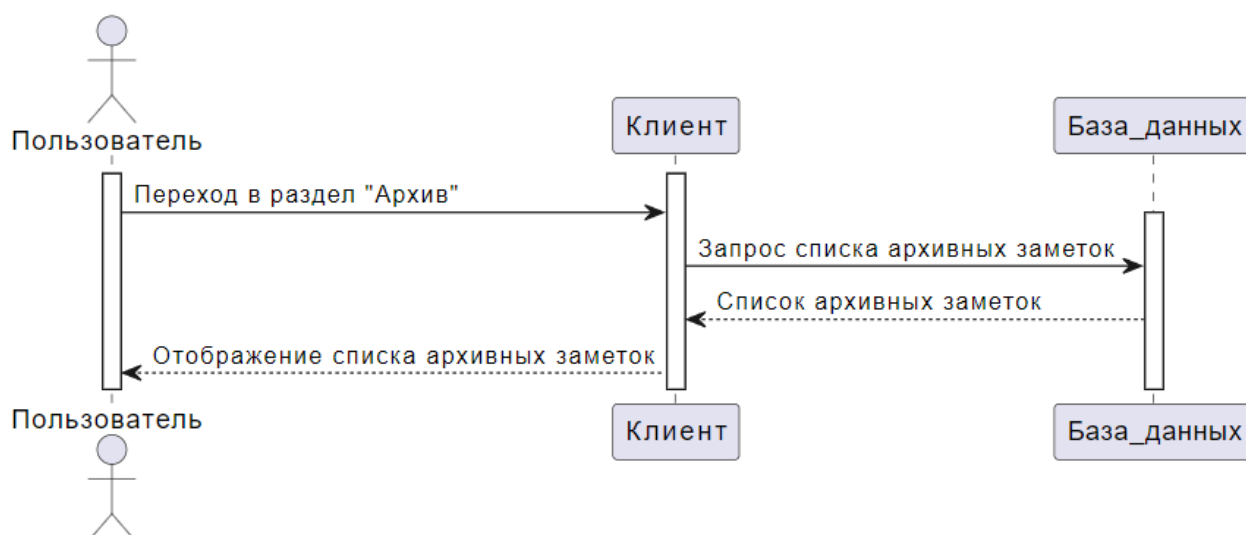


Рисунок 19 – Диаграмма последовательности «Отображение записей в разделе «Архив»

Функция `getDeleted` позволяет отобразить в разделе «Корзина» все заметки, которые находятся в корзине. Функция принимает на вход параметр с признаком нахождения заметки в корзине, при этом из таблицы выбираются записи, у которых этот параметр выставлен в «1».

Диаграмма последовательности приведена рисунке 20.

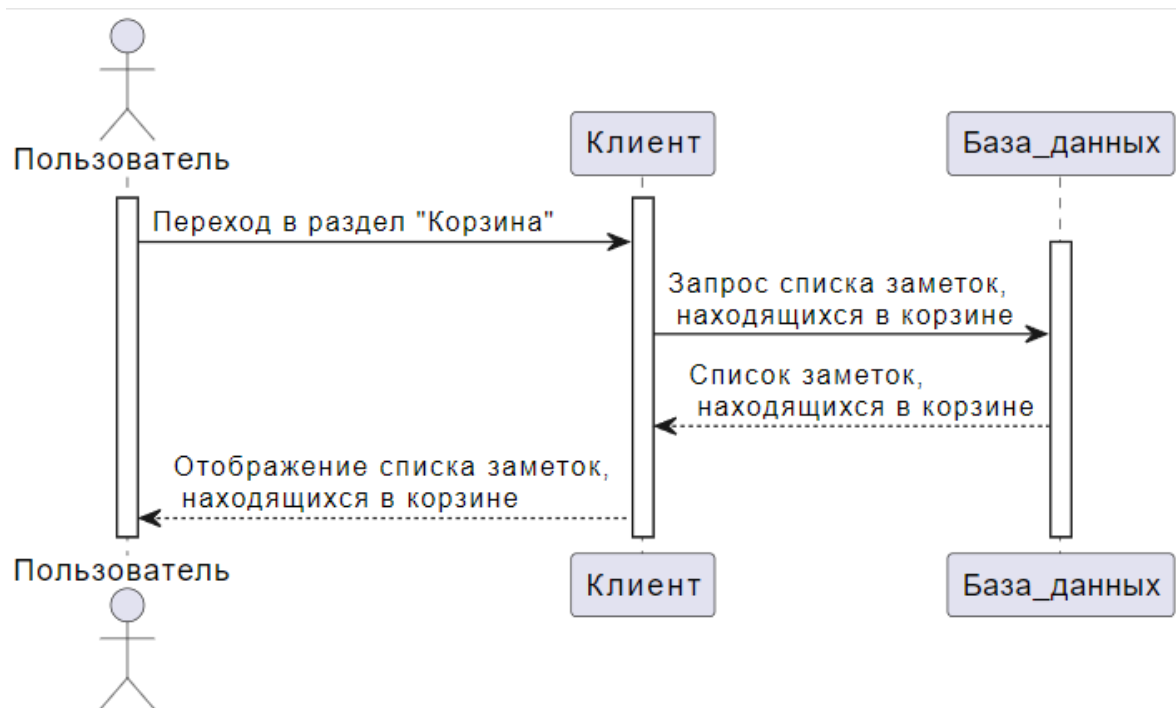


Рисунок 20 – Диаграмма последовательности «Отображение записей в разделе «Корзина»

1.8.3 Разработка экрана просмотра актуальных заметок

Для просмотра актуальных заметок создан виджет AllNotes.

По нажатию на кнопку таббара «Актуальное» будет открыта страница «Мои заметки», на которой будут отображены все заметки, не находящиеся в архиве или корзине.

Название страницы располагается в аппбаре. Если пользователь не создал ни одной актуальной заметки, на странице будет отображена надпись: «Вы пока не создали ни одной заметки». Отображение заметки реализовано с помощью виджета Card. Каждая карточка отображает заголовок, текст заметки, дату создания, тег и панель кнопок. Панель включает в себя три кнопки типа IconButton для редактирования заметки, перемещения заметки в архив и корзину.

В правом нижнем углу страницы располагается кнопка типа FloatingActionButton для создания новой заметки.

В аппбаре также располагается кнопка типа IconButton «Фильтр» для сортировки заметок по тегам. По нажатию на кнопку открывается модальное окно со списком доступных тегов.

По нажатию на карточку заметки Card осуществляется переход на экран просмотра заметки.

По нажатию на кнопку типа IconButton «Редактировать» осуществляется переход на экран редактирования заметки.

По нажатию на кнопку типа IconButton «Архив» осуществляется перемещение заметки в архив, при этом отображается оповещение о перемещении заметки в соответствующий раздел.

По нажатию на кнопку типа IconButton «Корзина» осуществляется перемещение заметки в корзину, при этом отображается оповещение о перемещении заметки в соответствующий раздел.

По нажатию на кнопку типа FloatingActionButton «Создать заметку» осуществляется переход на экран создания новой заметки.

1.8.4 Разработка экрана просмотра заметок в архиве

Для просмотра заметок в архиве создан виджет ArchivedNotes.

По нажатию на кнопку таббара «Архив» будет открыта страница «Архив», на которой будут отображены все заметки, находящиеся в архиве.

Название страницы располагается в аппбаре. Если в архиве нет ни одной заметки, на странице будет отображена надпись: «В архиве пока нет ни одной заметки». Отображение заметки реализовано с помощью виджета Card. Каждая карточка отображает заголовок, текст заметки, дату создания, тег и панель кнопок. Панель включает в себя три кнопки типа IconButton для редактирования заметки, перемещения заметки в актуальное и корзину.

В аппбаре также располагается кнопка типа IconButton «Фильтр» для сортировки заметок по тегам. По нажатию на кнопку открывается модальное окно со списком доступных тегов.

По нажатию на карточку заметки Card осуществляется переход на экран просмотра заметки.

По нажатию на кнопку типа IconButton «Редактировать» осуществляется переход на экран редактирования заметки.

По нажатию на кнопку типа IconButton «Актуальное» осуществляется перемещение заметки в раздел «Актуальное», при этом отображается оповещение о перемещении заметки в соответствующий раздел.

По нажатию на кнопку типа IconButton «Корзина» осуществляется перемещение заметки в корзину, при этом отображается оповещение о перемещении заметки в соответствующий раздел.

1.8.5 Разработка экрана просмотра заметок в корзине

Для просмотра заметок в корзине создан виджет DeletedNotes.

По нажатию на кнопку таббара «Корзина» будет открыта страница «Недавно удаленные», на которой будут отображены все заметки, находящиеся в корзине.

Название страницы располагается в аппбаре. Если в корзине нет ни одной заметки, на странице будет отображена надпись: «В корзине пока нет ни одной заметки». Отображение заметки реализовано с помощью виджета Card. Каждая карточка отображает заголовок, текст заметки, дату создания, тег и панель кнопок. Панель включает в себя три кнопки типа IconButton для перемещения заметки в актуальное, архив и корзину.

В аппбаре также располагается кнопка типа IconButton «Фильтр» для сортировки заметок по тегам. По нажатию на кнопку открывается модальное окно со списком доступных тегов.

По нажатию на карточку заметки Card осуществляется переход на экран просмотра заметки.

По нажатию на кнопку типа IconButton «Актуальное» осуществляется перемещение заметки в раздел «Актуальное», при этом отображается оповещение о перемещении заметки в соответствующий раздел.

По нажатию на кнопку типа IconButton «Архив» осуществляется перемещение заметки в архив, при этом отображается оповещение о перемещении заметки в соответствующий раздел.

По нажатию на кнопку типа IconButton «Корзина» вызывается диалоговое окно AlertDialog «Удаление заметки» с кнопками «Да» и «Отмена». По нажатию на кнопку «Да» осуществляется удаление заметки из базы данных, заметка перестает отображаться в разделе «Корзина».

1.8.6 Разработка экрана для создания заметки

Для создания новой заметки создан виджет AddNote.

По нажатию на кнопку типа FloatingActionButton «Создать заметку» осуществляется переход на экран создания новой заметки «Добавление новой заметки».

Название страницы располагается в аппбаре. В аппбаре также располагается кнопка типа IconButton для назначения тега заметки. При нажатии на кнопку «Тег» открывается модальное окно showModalBottomSheet со списком доступных для выбора тегов. Основной экран содержит два текстовых редактируемых поля TextField: заголовок, текст заметки, а также кнопку типа FloatingActionButton «Сохранить» и кнопку «Назад».

По нажатию на кнопку «Сохранить» введенные данные сохраняются в соответствующие поля таблицы all_notes, осуществляется переход на страницу «Мои заметки». Добавленная запись будет отображена в списке заметок в разделе «Актуальное».

При попытке создать заметку с пустым полем «Заголовок» или «Текст заметки» осуществляется вызов диалогового окна AlertDialog «Необходимо заполнить все поля!» с кнопкой «Ок».

1.8.7 Разработка экрана для просмотра заметки

Для просмотра существующей заметки создан виджет DisplayNote.

По нажатию на заметку будет открыта страница для просмотра полного текста заметки. На странице будет отображен заголовок заметки и основной текст заметки. Заголовок заметки имеет жирное начертание шрифта.

1.8.8 Разработка экрана для редактирования заметки

Для редактирования существующей заметки создан виджет UpdateNote.

					ПГУ1.090301.24.001 ПЗ	Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		

По нажатию на кнопку типа IconButton «Редактировать» осуществляется переход на экран редактирования заметки «Редактирование заметки».

Название страницы располагается в аппбаре. В поле «Заголовок» и «Текст заметки» отображаются ранее введенные данные. На странице располагается кнопка типа FloatingActionButton «Сохранить» и кнопка «Назад». По нажатию на кнопку «Сохранить» обновленные данные перезаписываются в соответствующие поля таблицы all_notes, осуществляется переход на страницу «Мои заметки». Обновленная запись будет отображена в списке заметок в разделе «Актуальное».

1.8.9 Разработка нижнего меню

Виджет BottomNavigationBar позволяет настроить нижнюю панель, именуемую таббаром или меню. Таббар позволяет упростить навигацию в приложении.

Компонент BottomNavigationBar состоит из трех элементов – кнопка «Актуальное», кнопка «Архив», кнопка «Корзина».

С помощью таббара пользователь может перемещаться между экранами в любом порядке в любой момент времени просмотра созданных заметок.

Иконка раздела, в котором находится пользователь, будет выделена цветом.

Нажатие на кнопку «Актуальное» позволяет перейти на страницу «Мои заметки» и отобразить список всех актуальных заметок.

Нажатие на кнопку «Архив» позволяет перейти на страницу «Архив» и отобразить список всех заметок, находящихся в архиве.

Нажатие на кнопку «Корзина» позволяет перейти на страницу «Недавно удаленные» и отобразить список всех заметок, находящихся в корзине.

1.8.10 Главный файл программы

В Flutter экраны и страницы часто называют маршрутами, которые представляют собой виджеты.

					ПГУ1.090301.24.001 ПЗ	Лист
						41
Изм.	Лист	№ докум.	Подпись	Дата		

В главном файле программы описан маршрут к каждому экрану приложения, начальная домашняя страница, таббар приложения, который одинаков для трех основных страниц приложения, а также функция main, являющаяся точкой входа в приложение.

Диаграмма классов разработанного приложения показана на рисунке 21.

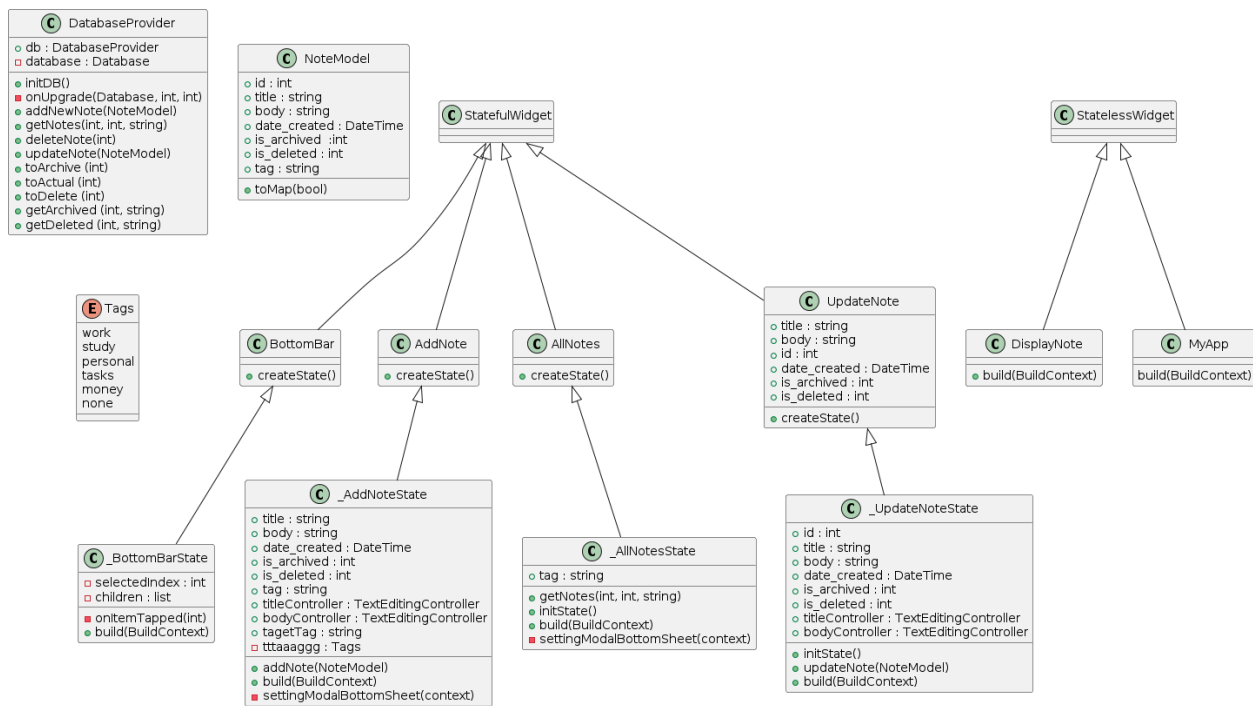


Рисунок 21 – Диаграмма классов

В процессе разработки архитектуры приложения были выявлены основные экранные формы и обозначены условия переходов между ними. На основе этих данных построена карта экранов, показанная на рисунке 22.

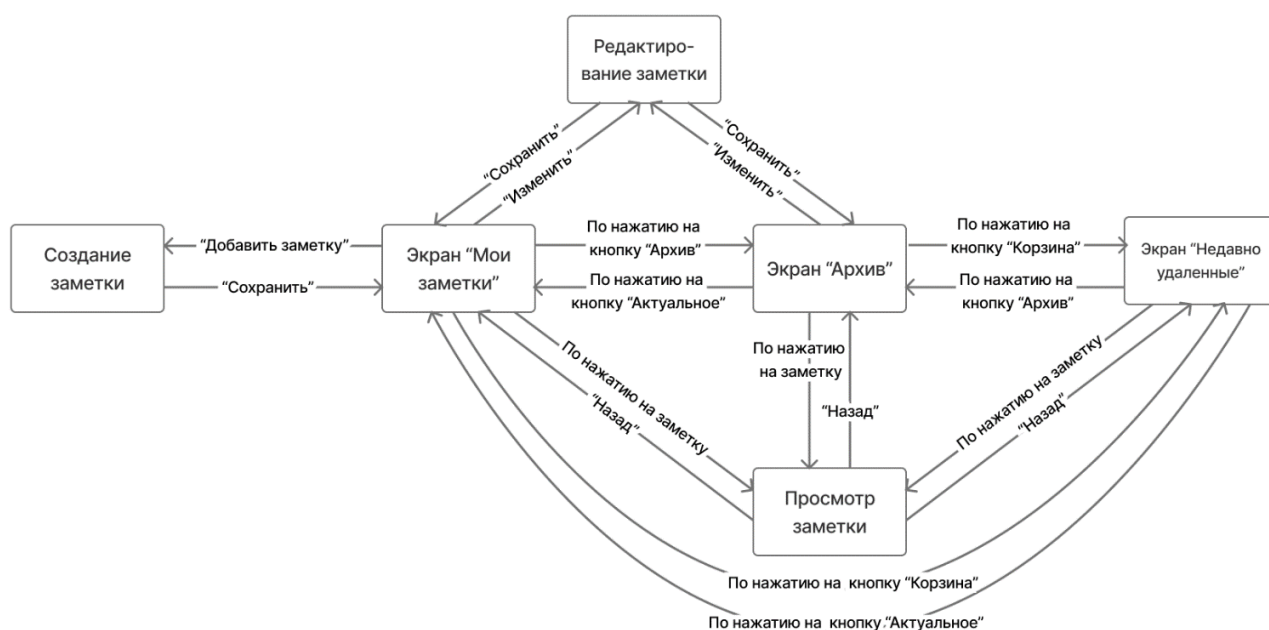


Рисунок 22 – Карта экранов

1.9 Тестирование и отладка приложения

Тестирование приложения является важным этапом разработки, поскольку оно позволяет убедиться в том, что приложение работает корректно и соответствует заявленным требованиям и потребностям пользователя.

Тестирование помогает выявить ошибки и дефекты в приложении до его выпуска и может помочь улучшить качество информационной системы, оптимизировать его работу и повысить удобство использования для пользователей.

Существует много видов тестирования, но основными являются ручное и автоматическое тестирование.

При ручном тестировании тестировщик вручную выполняет все шаги, чтобы проверить функционал приложения. Тестировщик имитирует реальные действия пользователя и старается охватить максимум функций продукта и найти ошибки.

При автоматизированном тестировании используются специальные инструменты для автоматического выполнения тестовых сценариев.

Автоматизированное тестирование – аналог ручного функционального тестирования, который выполняется программой-роботом, а не человеком.

Для тестирования разрабатываемого приложения выбрано ручное тестирование.

Список тест-кейсов:

- создание новой заметки;
- выход из процесса создания новой заметки;
- создание новой заметки, если не все поля заполнены;
- редактирование ранее созданной заметки;
- выход из процесса редактирования заметки;
- просмотр ранее созданной заметки;
- удаление заметки;
- просмотр всех заметок в разделе «Актуальное»;
- просмотр всех заметок в разделе «Архив»;
- просмотр всех заметок в разделе «Корзина»;
- перемещение заметки в актуальное;
- перемещение заметки в архив;
- перемещение заметки в корзину;
- фильтрация заметок по тегам.

1.9.1 Тест-кейс «Создание новой заметки»

Описание: тест-кейс предназначен для проверки корректности процесса создания новой заметки и записи информации в базу данных.

Требуемые условия для выполнения тестирования:

- пользователь находится в разделе «Актуальное»;
- если ранее пользователем не было создано ни одной заметки, на странице «Мои заметки» отображается сообщение «Вы пока не создали ни одной заметки»;
- если пользователем ранее были созданы заметки, они отображаются в виде списка на странице «Мои заметки».

Таблица 2 – Тест-кейс «Создание новой заметки»

№ шага	Описание шага	Результат
1	Нажать на кнопку «Создать заметку» в правом нижнем углу в разделе «Актуальное» (на странице «Мои заметки»).	Осуществляется переход на страницу «Добавление заметки». На странице «Добавление заметки» отображаются кнопки «Назад», «Сохранить», «Добавить тег» и текстовые поля для ввода заголовка заметки и основного текста заметки.
2	Ввести заголовок заметки и основной текст заметки, назначить заметке тег.	Введенный пользователем текст отображен в соответствующих полях.
3	Нажать на кнопку «Сохранить».	Введенные данные сохранены и записаны в базу данных. Осуществляется переход на страницу «Мои заметки». Система отображает на странице «Мои заметки» в разделе «Актуальное» только что созданную пользователем заметку.

Результат тестирования приведен на рисунках 23-27.

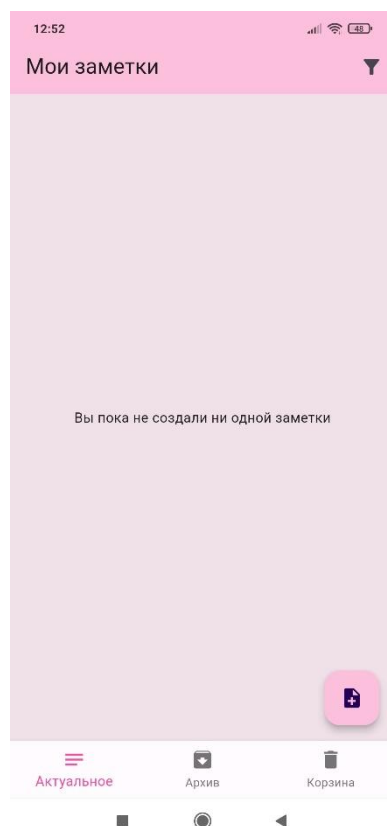


Рисунок 23 – Страница «Мои заметки», если не создано ни одной заметки

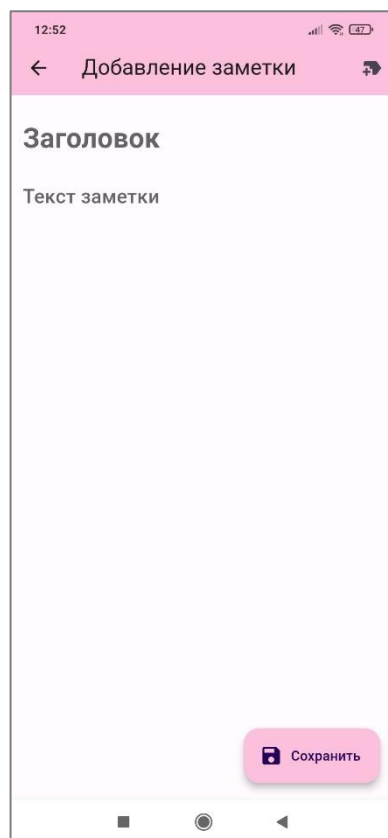


Рисунок 24 – Страница «Добавление заметки»

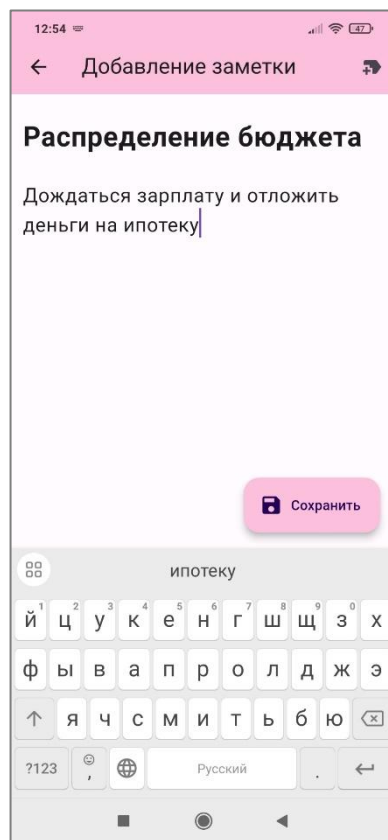


Рисунок 25 – Страница «Добавление заметки» с заполненными полями

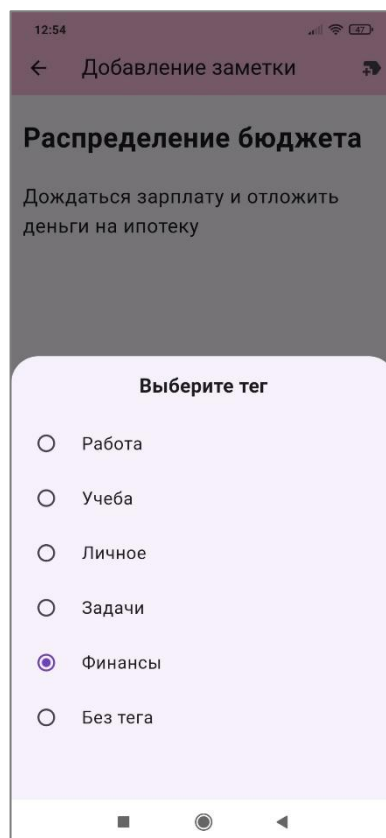


Рисунок 26 – Модальное окно «Выберите тег» для назначения тега заметке

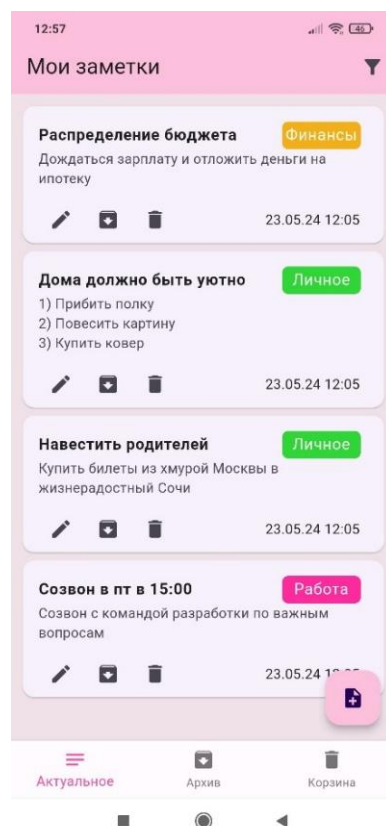


Рисунок 27 – Просмотр созданной заметки

Диаграмма деятельности для процесса создания новой заметки приведена на рисунке 28.

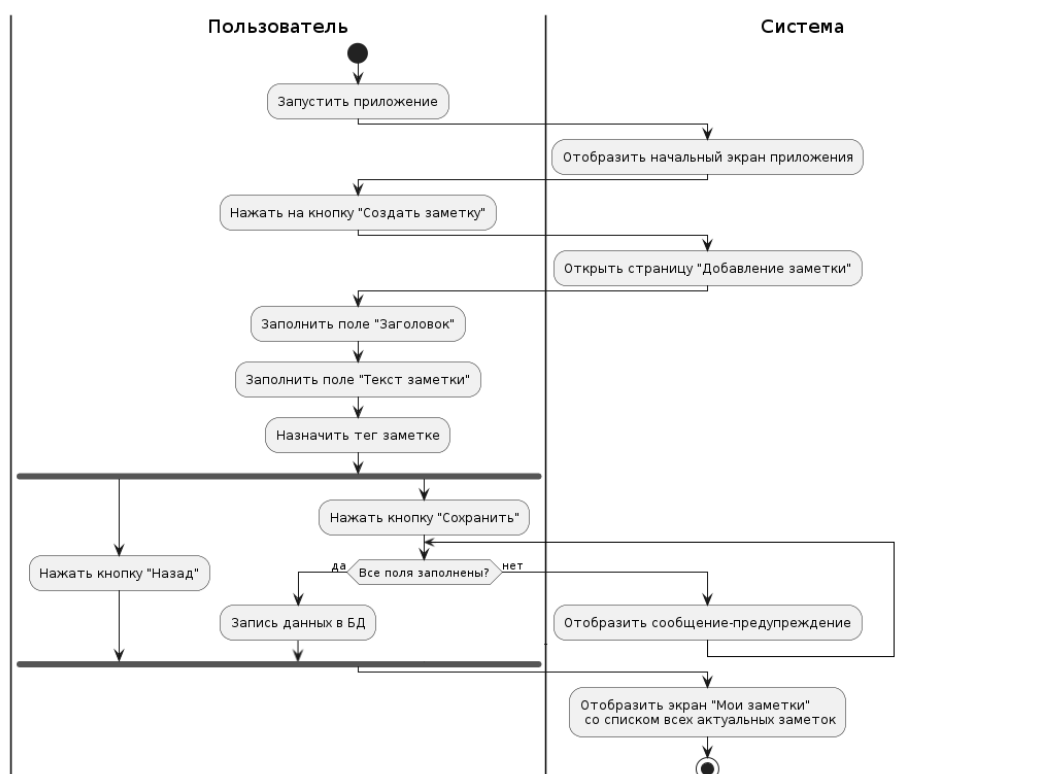


Рисунок 28 – Диаграмма деятельности "Создание заметки"

1.9.2 Тест-кейс «Выход из процесса создания новой заметки»

Описание: тест-кейс предназначен для проверки корректности выхода из процесса создания новой заметки.

Требуемое условие для выполнения тестирования: пользователь находится на странице «Добавление заметки».

Таблица 3 – Тест-кейс «Выход из процесса создания новой заметки»

№ шага	Описание шага	Результат
1	Нажать на кнопку «Назад» в верхнем левом углу экрана.	Осуществляется закрытие страницы «Добавление заметки» и переход на страницу «Мои заметки». Внесенные данные не сохранены, в разделе «Актуальное» новая заметка не отображается.

Результат тестирования приведен на рисунках 29-31.

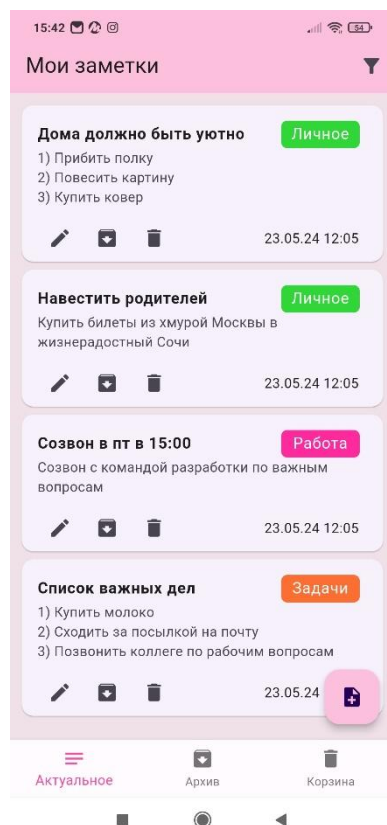


Рисунок 29 – Страница «Мои заметки» до создания новой заметки

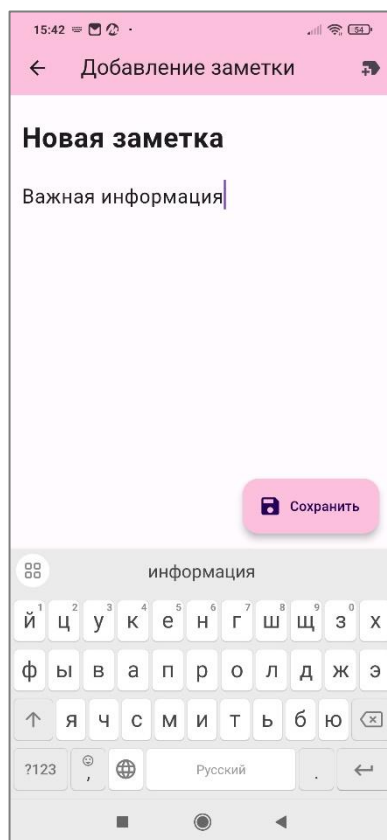


Рисунок 30 – Страница «Добавление заметки»

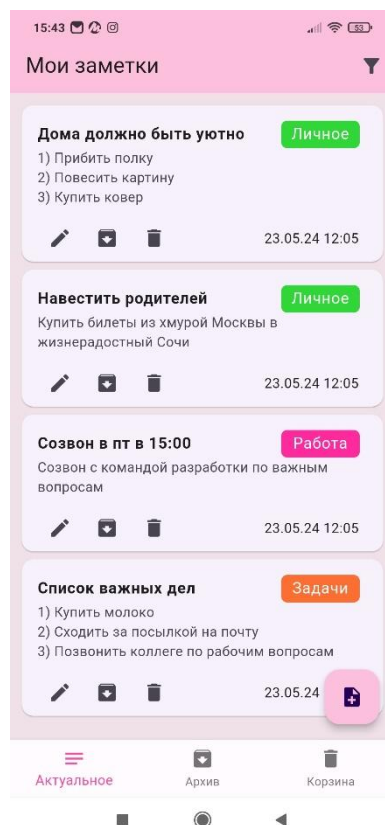


Рисунок 31 – Страница «Мои заметки» после выхода из процесса создания заметки

1.9.3 Тест-кейс «Создание новой заметки, если не все поля заполнены»

Описание: тест-кейс предназначен для проверки попытки сохранения заметки при ее создании, если не все поля заполнены.

Требуемые условия для выполнения тестирования:

- пользователь находится на странице «Добавление заметки»;
- поле «Заголовок» и/или поле «Текст заметки» не заполнено.

Таблица 4 – Тест-кейс «Создание новой заметки, если не все поля заполнены»

№ шага	Описание шага	Результат
1	Нажать на кнопку «Сохранить».	Отображается диалоговое окно «Необходимо заполнить все поля!» с кнопкой «Ок».
2	Нажать на кнопку «Ок».	Диалоговое окно закрывается. Отображается экран «Добавление заметки». Новая заметка не будет создана, запись

№ шага	Описание шага	Результат
		в базу данных не осуществлена.

Результат тестирования приведен на рисунке 32.

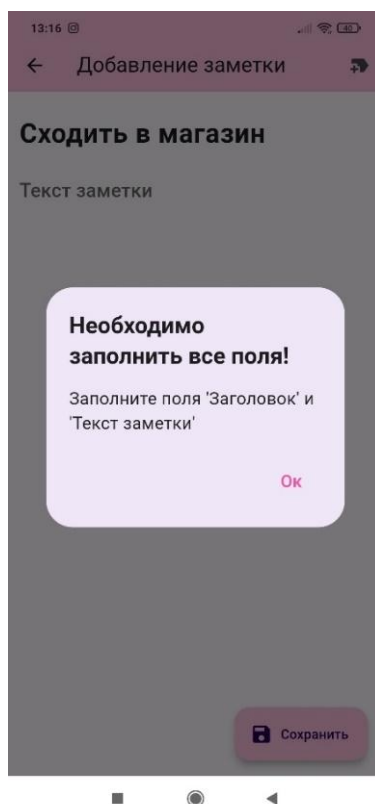


Рисунок 32 – Модальное окно «Необходимо заполнить все поля!»

1.9.4 Тест-кейс «Редактирование ранее созданной заметки»

Описание: тест-кейс предназначен для проверки процесса редактирования ранее созданной заметки.

Требуемое условие для выполнения тестирования: пользователь находится в разделе «Актуальное» или «Архив».

Таблица 5 – Тест-кейс «Редактирование ранее созданной заметки»

№ шага	Описание шага	Результат
1	Нажать на кнопку «Редактировать» на карточке заметки.	Отображается страница «Редактирование заметки» с кнопками «Назад» и «Сохранить». При этом в полях «Заголовок» и «Текст заметки» будет отображен старый текст заметки, введенный пользователем ранее.

№ шага	Описание шага	Результат
2	Внести правки в поле «Заголовок» и/или «Текст заметки».	В поле «Заголовок» и/или «Текст заметки» отображается измененный текст.
3	Нажать на кнопку «Сохранить».	Внесенные изменения сохранены, запись в базе данных обновлена. Осуществляется закрытие страницы «Редактирование заметки» и осуществляется переход на страницу, с которой был совершен переход к редактированию заметки. На соответствующей странице будет отображена карточка заметки с обновленными данными.

Результат тестирования приведен на рисунках 33-35.

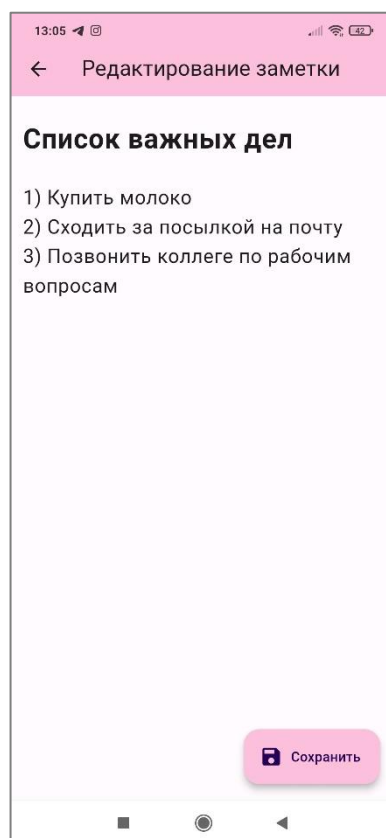


Рисунок 33 – Страница «Редактирование заметки» до внесения изменений

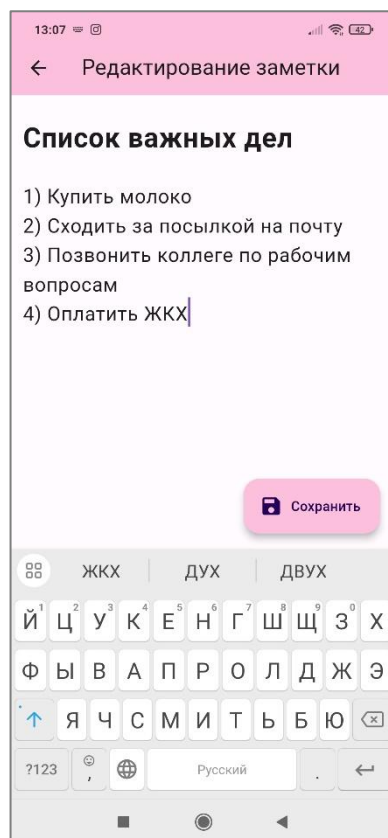


Рисунок 34 – Страница «Редактирование заметки» после внесения изменений

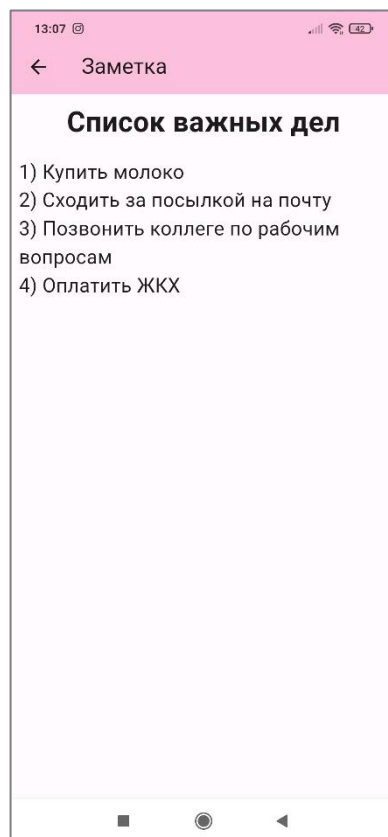


Рисунок 35 – Результат редактирования заметки

1.9.5 Тест-кейс «Выход из процесса редактирования заметки»

Описание: тест-кейс предназначен для проверки корректности выхода из процесса редактирования заметки.

Требуемое условие для выполнения тестирования: пользователь находится на странице «Редактирование заметки».

Таблица 6 – Тест-кейс «Выход из процесса редактирования заметки»

№ шага	Описание шага	Результат
1	Нажать на кнопку «Назад» в верхнем левом углу экрана.	Осуществляется закрытие страницы «Редактирование заметки» и осуществляется переход на страницу, с которой был совершен переход к редактированию заметки. Внесенные данные не сохранены.

Результат тестирования приведен на рисунках 36-38.

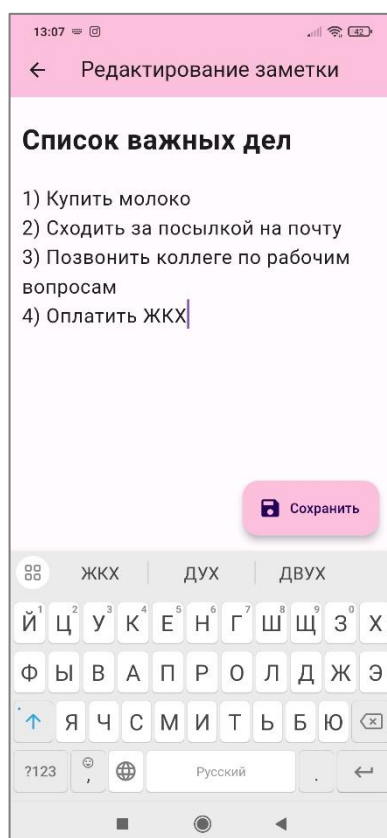


Рисунок 36 – Страница «Редактирование заметки»

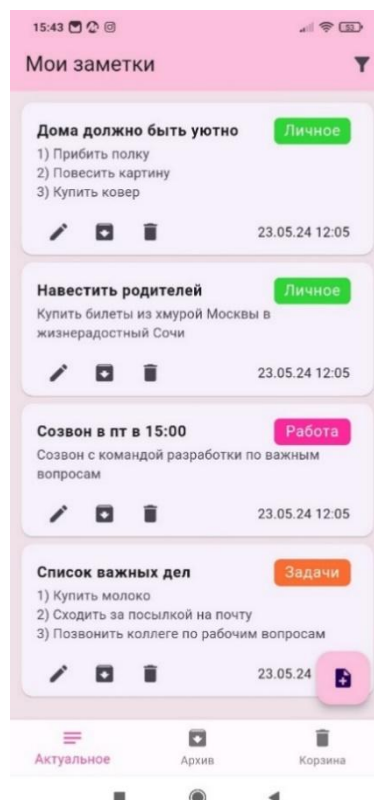


Рисунок 37 – Страница «Мои заметки» при выходе из процесса редактирования без сохранения внесенных изменений

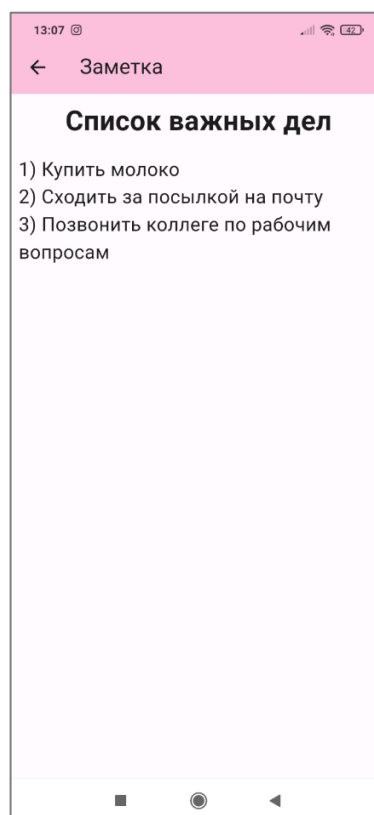


Рисунок 38 – Страница просмотра заметки после выхода из процесса редактирования без сохранения внесенных изменений

1.9.6 Тест-кейс «Просмотр ранее созданной заметки»

Описание: тест-кейс предназначен для проверки корректности отображения данных ранее созданной заметки.

Требуемое условие для выполнения тестирования: пользователь находится в разделе «Актуальное», «Архив» или «Корзина».

Таблица 7 – Тест-кейс «Просмотр ранее созданной заметки»

№ шага	Описание шага	Результат
1	Нажать на карточку заметки.	Переход к странице «Заметка» с кнопкой «Назад».
2	Просмотреть заметку.	В полях «Заголовок» и «Текст заметки» будет отображен текст заметки, введенный пользователем ранее.
3	Нажать на кнопку «Назад».	Осуществляется закрытие страницы «Заметка» и возврат на предыдущую страницу, с которой был совершен переход к просмотру заметки.

Результат тестирования приведен на рисунках 39-40.

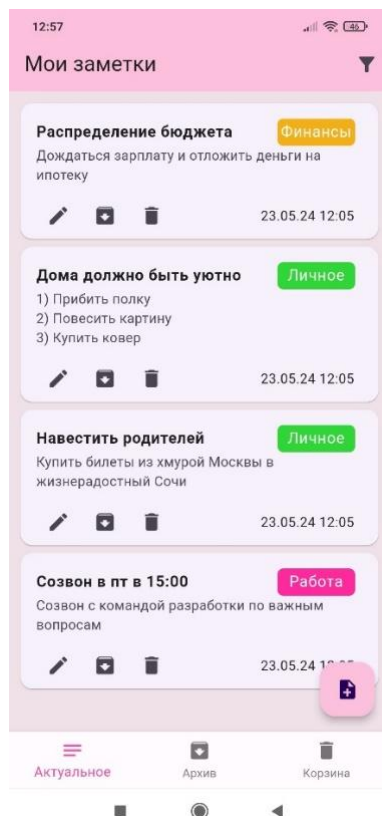


Рисунок 39 – Страница «Мои заметки»

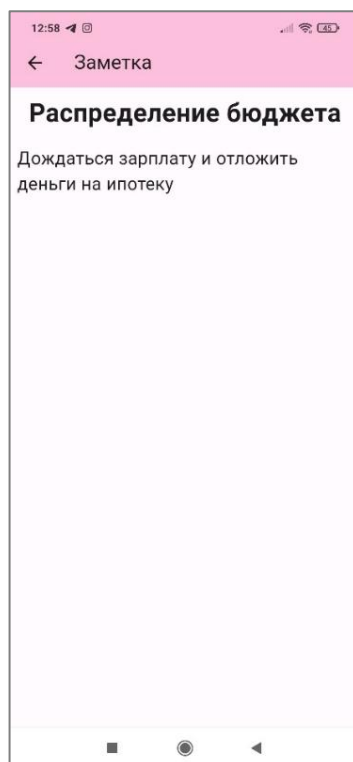


Рисунок 40 – Страница просмотра заметки

Диаграмма деятельности для процесса просмотра заметки приведена на рисунке 41.

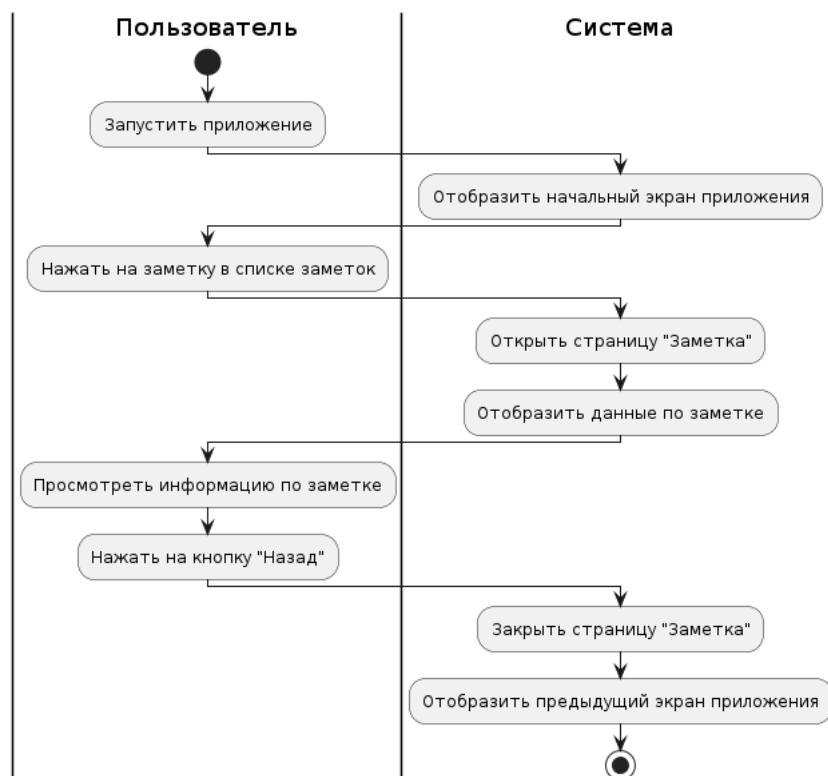


Рисунок 41 – Диаграмма деятельности "Просмотр заметки"

1.9.7 Тест-кейс «Удаление заметки»

Описание: тест-кейс предназначен для проверки процесса удаления заметки.

Требуемые условия для выполнения тестирования:

- пользователь находится в разделе «Корзина»;
- удаляемая заметка находится в разделе «Корзина».

Таблица 8 – Тест-кейс «Удаление заметки»

№ шага	Описание шага	Результат
1	Нажать на кнопку «Удалить заметку».	Осуществляется вызов диалогового окна «Удаление заметки» с кнопками «Да» и «Отмена».
2	Нажать на кнопку «Да».	Осуществляется удаление заметки из базы данных. После успешного удаления заметки, выдается сообщение «Заметка удалена». Заметка не отображается в разделе «Корзина».

Результат тестирования приведен на рисунках 42-43.

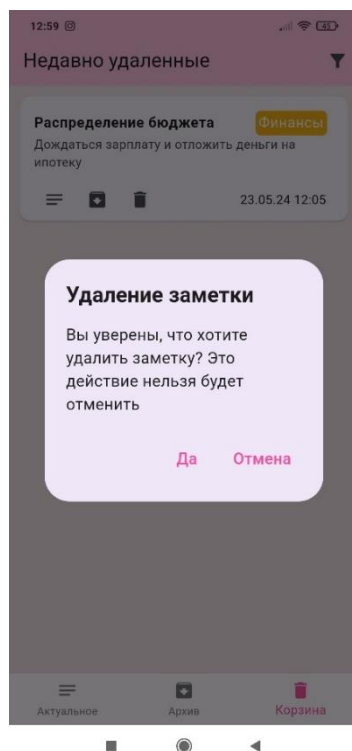


Рисунок 42 – Модальное окно «Удаление заметки»

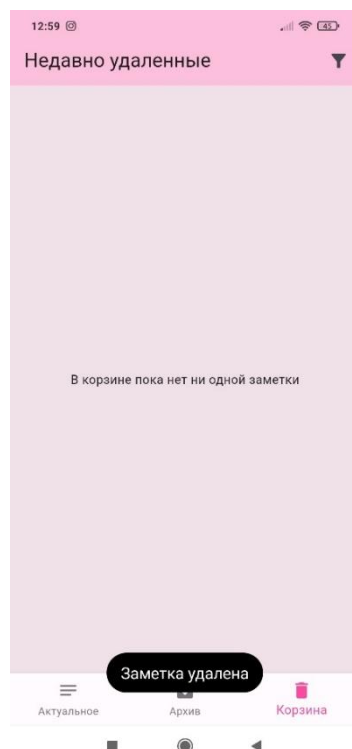


Рисунок 43 – Информационное сообщение «Заметка удалена»

Диаграмма деятельности для процесса удаления заметки приведена на рисунке 44.

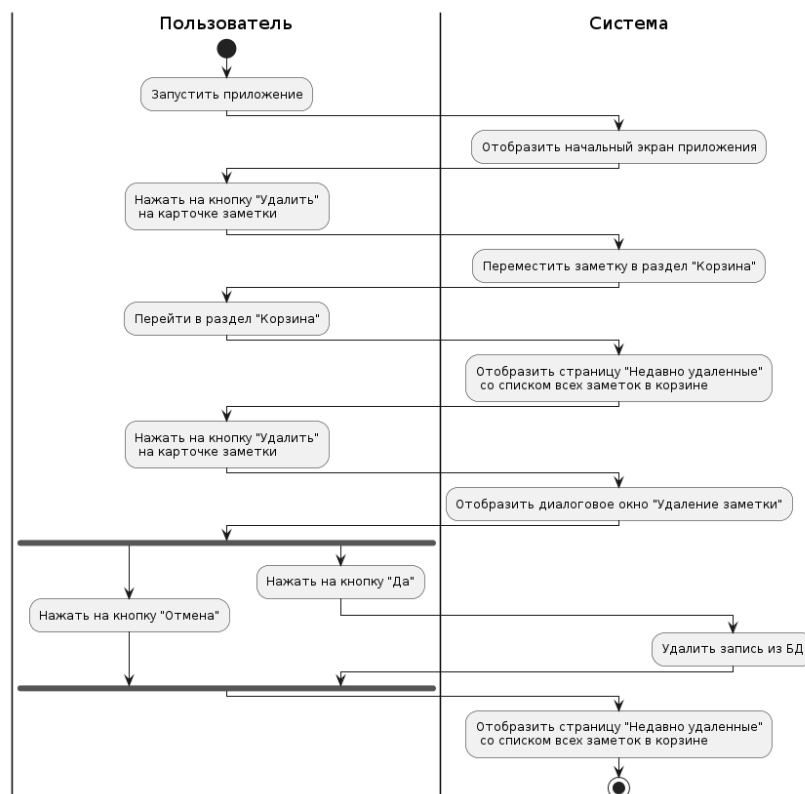


Рисунок 44 – Диаграмма деятельности «Удаление заметки»

1.9.8 Тест-кейс «Просмотр всех заметок в разделе «Актуальное»

Описание: тест-кейс предназначен для проверки корректности отображения списка заметок в разделе «Актуальное».

Требуемое условие для выполнения тестирования: пользователь запускает приложение или находится в разделе «Архив»/«Корзина».

Таблица 9 – Тест-кейс «Просмотр всех заметок в разделе «Актуальное»

№ шага	Описание шага	Результат
1	Нажать на кнопку «Актуальное» на таббаре приложения при переходе из раздела «Архив»/«Корзина», иначе запустить приложение.	Осуществляется открытие раздела «Актуальное». Если заметки в разделе «Актуальное» отсутствуют, отображается сообщение «Вы пока не создали ни одной заметки». В случае, если в базе данных имеются записи, то в разделе «Актуальное» отображается список созданных заметок. Каждая заметка представляет собой карточку, содержащую заголовок, основной текст заметки, дату создания заметки, тег и кнопки для управления заметкой: кнопку «Редактировать», кнопку «Переместить в архив», «Переместить в корзину».

Результат тестирования приведен на рисунках 45-46.

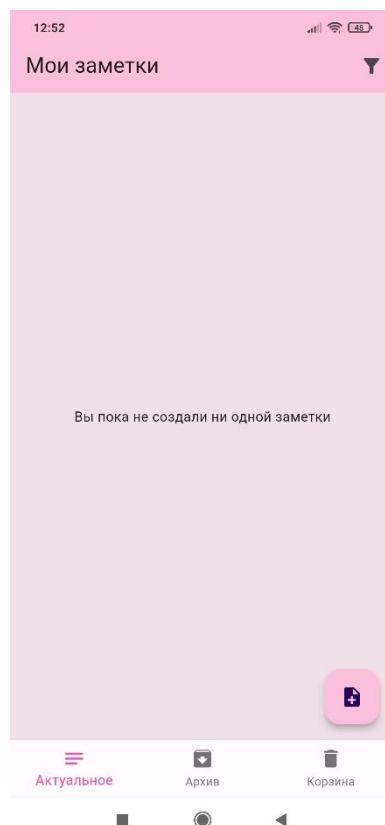


Рисунок 45 – Раздел «Актуальное», если ни одна заметка не создана

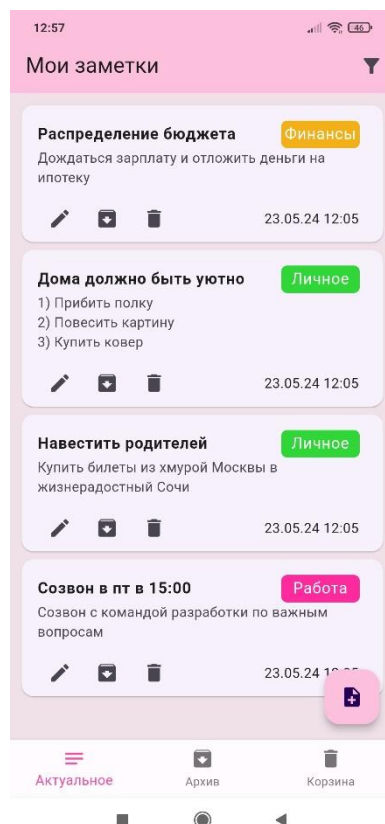


Рисунок 46 – Просмотр заметок в разделе «Актуальное»

					ПГУ1.090301.24.001 ПЗ	Лист
						61
Изм.	Лист	№ докум.	Подпись	Дата		

1.9.9 Тест-кейс «Просмотр всех заметок в разделе «Архив»

Описание: тест-кейс предназначен для проверки корректности отображения списка заметок в разделе «Архив».

Требуемое условие для выполнения тестирования: пользователь находится в разделе «Актуальное»/«Корзина».

Таблица 10 – Тест-кейс «Просмотр всех заметок в разделе «Архив»

№ шага	Описание шага	Результат
1	Нажать на кнопку «Архив» на таббаре приложения при переходе из раздела «Актуальное»/«Корзина».	Осуществляется открытие раздела «Архив». Если заметки в разделе «Архив» отсутствуют, отображается сообщение «В архиве пока нет ни одной заметки». В случае, если в базе данных имеются записи с признаком is_archived=1, то в разделе «Архив» отображается список архивных заметок. Каждая заметка представляет собой карточку, содержащую заголовок, основной текст заметки, дату создания заметки, тег и кнопки для управления заметкой: кнопку «Редактировать», кнопку «Переместить в актуальное», «Переместить в корзину».

Результат тестирования приведен на рисунках 47-48.



Рисунок 47 – Раздел «Архив», если в нем нет ни одной заметки

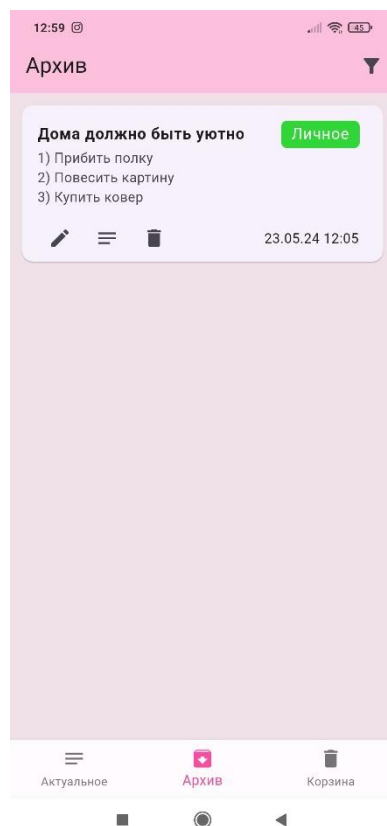


Рисунок 48 – Просмотр заметок в разделе «Архив»

					ПГУ1.090301.24.001 ПЗ	Лист
						63
Изм.	Лист	№ докум.	Подпись	Дата		

1.9.10 Тест-кейс «Просмотр всех заметок в разделе «Корзина»

Описание: тест-кейс предназначен для проверки корректности отображения списка заметок в разделе «Корзина».

Требуемое условие для выполнения тестирования: пользователь находится в разделе «Актуальное»/«Архив».

Таблица 11 – Тест-кейс «Просмотр всех заметок в разделе «Корзина»

№ шага	Описание шага	Результат
1	Нажать на кнопку «Корзина» на таббаре приложения при переходе из раздела «Актуальное»/«Архив».	Осуществляется открытие раздела «Корзина». Если заметки в разделе «Корзина» отсутствуют, отображается сообщение «В корзине пока нет ни одной заметки». В случае, если в базе данных имеются записи с признаком is_deleted=1, то в разделе «Корзина» отображается список заметок, перемещенных в корзину. Каждая заметка представляет собой карточку, содержащую заголовок, основной текст заметки, дату создания заметки, тег и кнопки для управления заметкой: кнопку «Переместить в актуальное», кнопку «Переместить в архив», «Удалить».

Результат тестирования приведен на рисунках 49-50.



Рисунок 49 – Раздел «Корзина», если в нем нет ни одной заметки

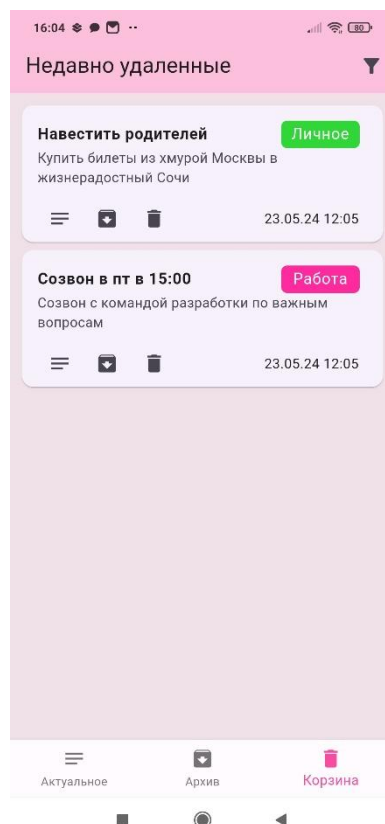


Рисунок 50 – Просмотр заметок в разделе «Корзина»

1.9.11 Тест-кейс «Перемещение заметки в актуальное»

Описание: тест-кейс предназначен для проверки процесса перемещения заметки в раздел «Актуальное».

Требуемое условие для выполнения тестирования: пользователь находится в разделе «Архив» или «Корзина».

Таблица 12 – Тест-кейс «Перемещение заметки в актуальное»

№ шага	Описание шага	Результат
1	Нажать на кнопку «Переместить в актуальное» на карточке заметки.	Осуществляется перемещение заметки в раздел «Актуальное». После успешного перемещения заметки, система выдаст сообщение «Заметка перемещена в актуальное».

Результат тестирования приведен на рисунках 51.

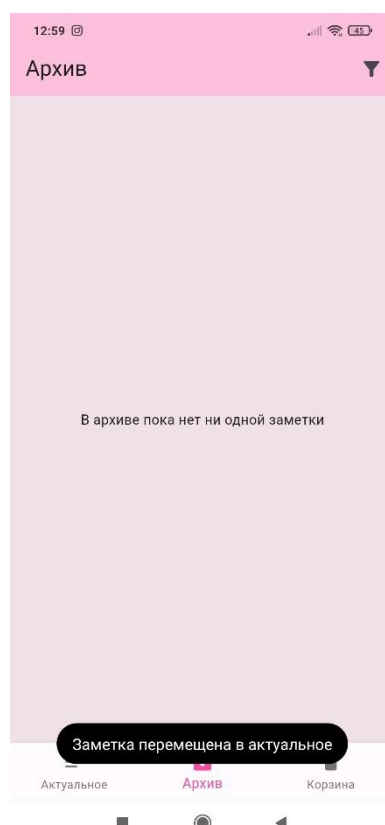


Рисунок 51 – Результат перемещения заметки в раздел «Актуальное»

1.9.12 Тест-кейс «Перемещение заметки в архив»

Описание: тест-кейс предназначен для проверки процесса перемещения заметки в раздел «Архив».

					ПГУ1.090301.24.001 ПЗ	Лист
						66
Изм.	Лист	№ докум.	Подпись	Дата		

Требуемое условие для выполнения тестирования: пользователь находится в разделе «Актуальное» или «Корзина».

Таблица 13 – Тест-кейс «Перемещение заметки в архив»

№ шага	Описание шага	Результат
1	Нажать на кнопку «Переместить в архив» на карточке заметки.	Осуществляется перемещение заметки в раздел «Архив». После успешного перемещения заметки, система выдаст сообщение «Заметка перемещена в архив».

Результат тестирования приведен на рисунках 52.

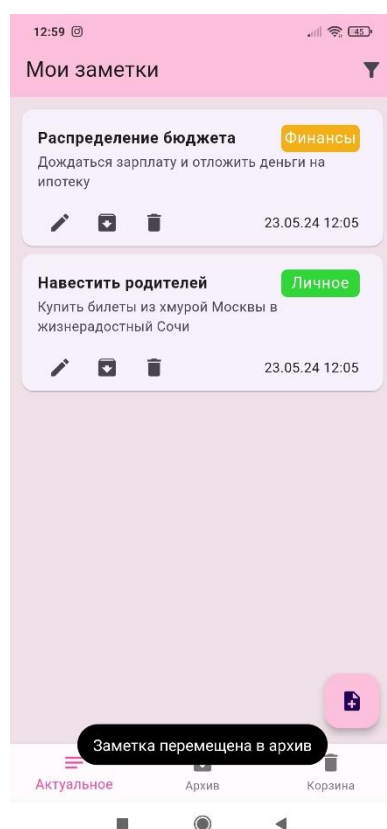


Рисунок 52 – Результат перемещения заметки в раздел «Архив»

Диаграмма деятельности для процесса перемещения заметки в раздел «Архив» приведена на рисунке 53.

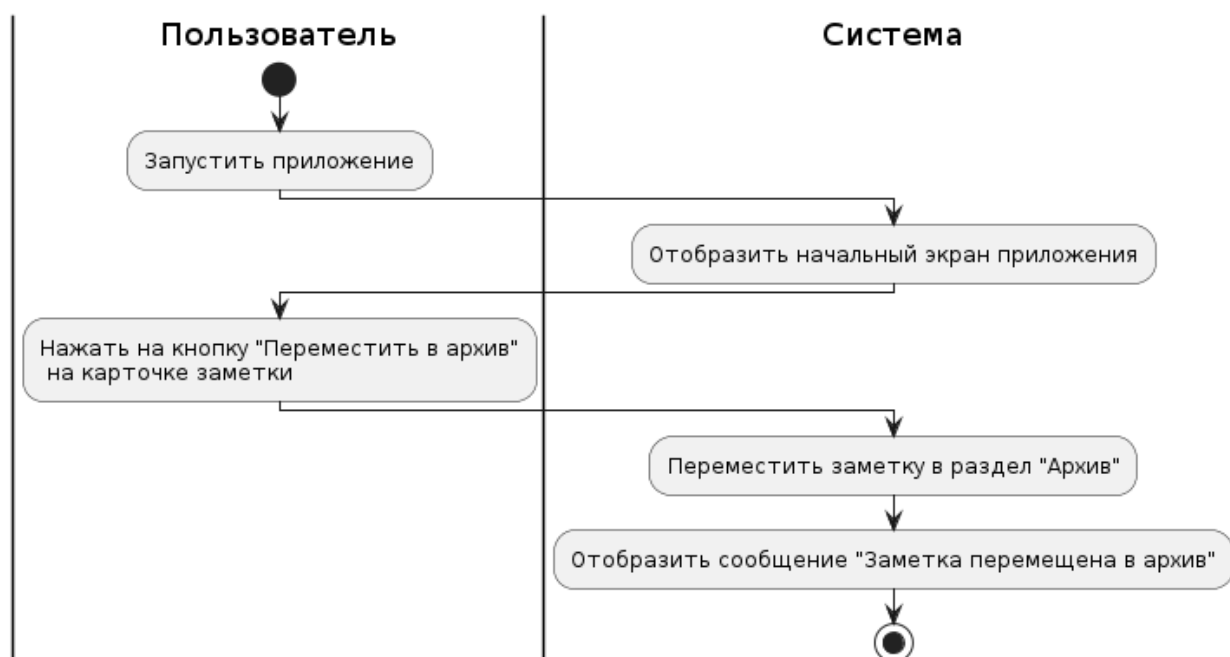


Рисунок 53 – Диаграмма деятельности «Перемещение заметки в архив»

1.9.13 Сценарий «Перемещение заметки в корзину»

Описание: тест-кейс предназначен для проверки процесса перемещения заметки в раздел «Корзина».

Требуемое условие для выполнения тестирования: пользователь находится в разделе «Актуальное» или «Архив».

Таблица 14 – Тест-кейс «Перемещение заметки в корзину»

№ шага	Описание шага	Результат
1	Нажать на кнопку «Переместить в корзину» на карточке заметки.	Осуществляется перемещение заметки в раздел «Корзина». После успешного перемещения заметки, система выдаст сообщение «Заметка перемещена в корзину».

Результат тестирования приведен на рисунках 54.

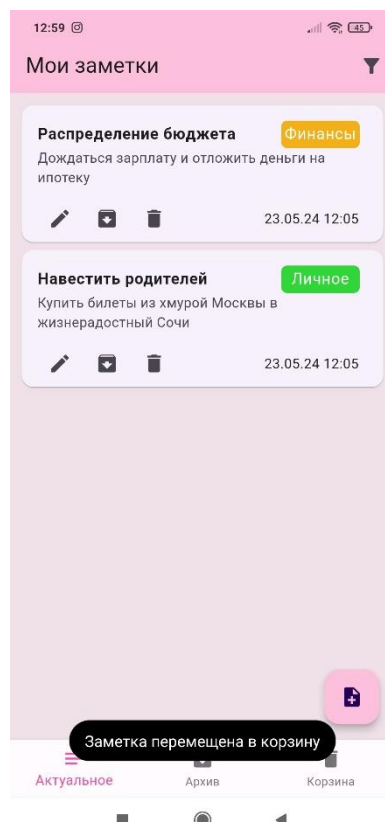


Рисунок 54 – Результат перемещения заметки в раздел «Корзина»

1.9.14 Тест-кейс «Фильтрация заметок по тегам»

Описание: тест-кейс предназначен для проверки процесса фильтрации заметок по тегам.

Требуемое условие для выполнения тестирования: пользователь находится в разделе «Актуальное», «Архив» или «Корзина».

Таблица 15 – Тест-кейс «Фильтрация заметок по тегам»

№ шага	Описание шага	Результат
1	Нажать на кнопку «Фильтр» в аппбаре страницы.	Открывается модальное окно «Выберите фильтр» со списком доступных фильтров.
2	Нажать на фильтр в списке	На странице отображаются все заметки с назначенным тегом, который был выбран в качестве фильтра. Иначе, если нет заметок с выбранным тегом, ни одна из заметок не отображается.

Результат тестирования приведен на рисунках 55-57.

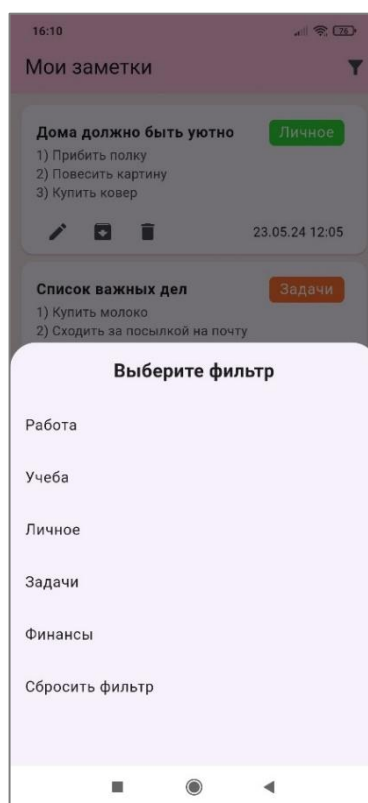


Рисунок 55 – Модальное окно «Выберите фильтр» для фильтрации заметок по тегам

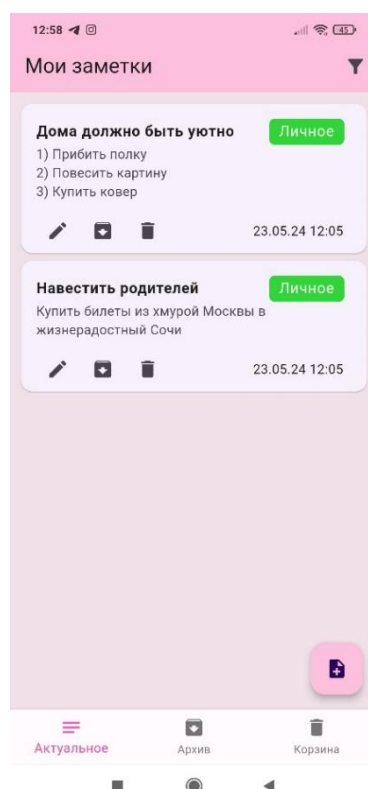


Рисунок 56 – Результат фильтрации заметок по тегу «Личное»

					ПГУ1.090301.24.001 ПЗ	Лист
						70
Изм.	Лист	№ докум.	Подпись	Дата		

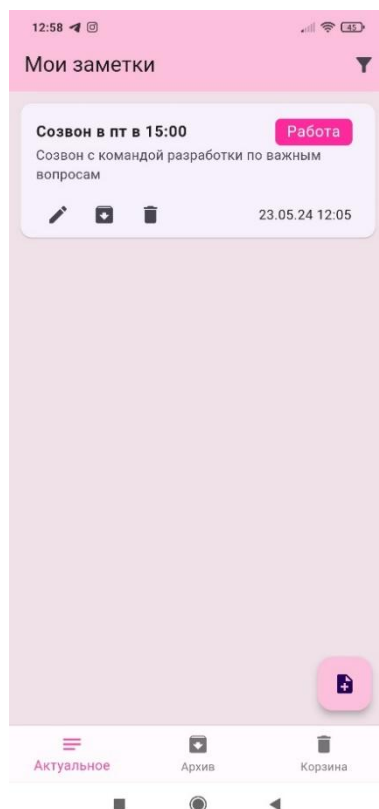


Рисунок 57 – Результат фильтрации заметок по тегу «Работа»

					ПГУ1.090301.24.001 ПЗ	Лист
						71
Изм.	Лист	№ докум.	Подпись	Дата		

2 Безопасность жизнедеятельности

На всех этапах разработки проекта необходимо гарантировать безопасность жизнедеятельности и труда работников. Для обеспечения безопасных условий труда необходимо проанализировать риски, связанные с производственной средой, физической и умственной нагрузкой, а также стрессом. Поскольку негативные факторы могут оказывать отрицательное воздействие на организм человека, важно учитывать их и предпринимать необходимые меры для минимизации их влияния.

При работе на компьютере человек сталкивается с несколькими негативными факторами:

- наличие огромной отрицательной нагрузки на зрение, которая может стать причиной его необратимого ухудшения, а также покраснений и синдрома «сухого глаза»;
- длительное пребывание в одной позе и ограниченная подвижность. Неправильная поза при работе за персональным компьютером способна вызвать разнообразные заболевания суставов, грудной клетки и периодические боли разного характера;
- чрезмерная нагрузка на психику. Требуется постоянная концентрация внимания на большом количестве деталей, которая может стать причиной переутомления.

2.1 Требования к рабочему месту

Рабочее место программиста и взаимное расположение всех его элементов должно соответствовать антропометрическим, физическим и психологическим требованиям. При организации рабочего места программиста должны быть соблюдены следующие основные условия: оптимальное размещение оборудования, входящего в состав рабочего места и достаточное рабочее пространство, позволяющее осуществлять все необходимые движения и перемещения.

					ПГУ1.090301.24.001 ПЗ	Лист
						72
Изм.	Лист	№ докум.	Подпись	Дата		

Главными элементами рабочего места программиста являются стол и кресло. Основным рабочим положением является положение сидя. Такая рабочая поза вызывает минимальное утомление программиста.

Для комфортной работы стол должен удовлетворять следующим условиям:

- высота стола должна быть выбрана с учетом возможности сидеть свободно, в удобной позе, при необходимости опираясь на подлокотники;
- нижняя часть стола должна быть сконструирована так, чтобы программист мог удобно сидеть, не был вынужден поджимать ноги;
- поверхность стола должна обладать свойствами, исключающими появление бликов в поле зрения программиста;
- конструкция стола должна предусматривать наличие выдвижных ящиков (не менее 3 для хранения документации, листингов, канцелярских принадлежностей);
- высота рабочей поверхности рекомендуется в пределах 680-760 мм. Высота поверхности, на которую устанавливается клавиатура, должна быть около 650 мм;
- размер рабочей поверхности стола для ПЭВМ следует считать ширину 800, 1000, 1200 и 1400 мм и глубину 800 и 1000 мм при нерегулируемой его высоте, равной 725 мм.

Большое значение придается характеристикам рабочего кресла программиста. Так, рекомендуемая высота сиденья над уровнем пола находится в пределах 420-550 мм. Поверхность сиденья мягкая, ширина и глубина поверхности сиденья не менее 400 мм, передний край закругленный, а угол наклона спинки – регулируемый.

Немаловажное значение имеет положение экрана монитора при работе программиста. Также должна предусматриваться возможность регулирования экрана по высоте и наклону.

Во время пользования компьютером монитор следует устанавливать на расстоянии 50-60 см от глаз, верхняя часть видеодисплея должна быть на

					ПГУ1.090301.24.001 ПЗ	Лист
						73
Изм.	Лист	№ докум.	Подпись	Дата		

уровне глаз или чуть ниже. Если же экран установлен высоко, нарушается функция моргания, что приводит к быстрой утомляемости глаз.

2.2 Освещение рабочего места

Работа программиста происходит в помещениях, поэтому освещение в них должно приближаться к оптимальным условиям зрительного солнечного освещения.

Требования к освещенности в помещениях, где установлены компьютеры, следующие: при выполнении зрительных работ высокой точности общая освещенность должна составлять 300 лк, а комбинированная – 750 лк; аналогичные требования при выполнении работ средней точности – 200 и 300 лк соответственно. При этом монитор и источники света должны быть расположены таким образом, чтобы не создавать бликов на поверхности экрана. Кроме того, все поле зрения должно быть освещено достаточно равномерно.

Помещения с ЭВМ должны иметь и естественное, и искусственное освещение. Естественное освещение должно осуществляться через светопроемы, ориентированные преимущественно на север и северо-восток. Искусственное освещение в помещениях эксплуатации ЭВМ должно осуществляться системой общего равномерного освещения. В качестве источников искусственного освещения обычно используются люминесцентные лампы, которые попарно объединяются в светильники, которые должны располагаться над рабочими поверхностями равномерно.

Ограничить отраженную блескость на рабочих поверхностях можно за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране не должна превышать 40 кд/кв.м и яркость потока, при изменении системы отраженного освещения, не должна превышать 200 кд/кв.м.

					ПГУ1.090301.24.001 ПЗ	Лист
						74
Изм.	Лист	№ докум.	Подпись	Дата		

Поскольку экран монитора – это тоже источник света, при постоянном чтении информации с него, происходит быстрое утомление глаз, особенно если яркость свечения монитора установлена слишком высокой. Также раздражение глаз вызывает мерцание изображения на мониторе, вызванной низкой частотой кадровой развертки. В целях снижения мерцания экрана рекомендуется устанавливать частоту кадров не менее 75 Гц для ЭЛТ-мониторов, для ЖК-мониторов достаточной является минимальная частота кадров в 60 Гц.

2.3 Микроклимат рабочего помещения

Микроклимат рабочего помещения – это климат внутренней среды этого помещения, который определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха.

Средняя температура воздуха в помещении офиса должна составлять +22°C, относительная влажность – 46%, атмосферное давление – 750 мм.рт.ст., содержание пыли – не более 10 мг/м воздуха рабочего места, максимальные размеры частиц – 2 мкм.

Вычислительная техника является источником существенных тепловыделений, что может привести к повышению температуры и снижению относительной влажности в помещении. Для создания и поддержания оптимального микроклимата, независимо от наружных условий в холодное время года используется водяное отопление, в теплое время года применяется кондиционирование воздуха. Кондиционер, с помощью приборов автоматического регулирования поддерживает в помещении заданные параметры воздушной среды.

Температура воздуха в помещении офиса не должна опускаться ниже +19°C, а при полной загруженности оборудования температура воздуха в офисе не должна превышать +25°C.

					ПГУ1.090301.24.001 ПЗ	Лист
						75
Изм.	Лист	№ докум.	Подпись	Дата		

2.4 Правила при использовании мобильного устройства

Сотовые телефоны стали незаменимым атрибутом современного мира. С помощью них человек может оставаться на связи из любой точки мира. Но помимо благ мобильные устройства могут нести и негативные последствия при неправильном использовании.

Основными симптомами неблагоприятного воздействия сотового телефона на состояние здоровья являются:

- головные боли;
- сонливость;
- раздражительность;
- нарушения памяти и ухудшение способности к концентрации;
- синдром хронической усталости;
- боль и резь в глазах;
- сухость слизистой оболочки глаза;
- прогрессирующее ухудшение зрения;
- перепады артериального давления и пульса.

Основные правила безопасного использования смартфона:

- контроль за работой и состоянием смартфона. Если у смартфона возникли проблемы, такие как чрезмерное нагревание, неожиданная звуковая сигнализация, плохое отображение на дисплее, вздутие батареи или разбитый дисплей, необходимо обратиться за помощью к специалистам. Самостоятельно открывать или ремонтировать устройство может быть крайне опасно;
- использование только качественных зарядных устройств. Важно использовать только оригинальные зарядные устройства и кабели. Такие аксессуары обеспечивают безопасность и имеют встроенные механизмы защиты от неправильного использования. Некачественные зарядные устройства могут вызвать чрезмерный перегрев, возгорание или даже взрыв мобильного устройства;

– ограничение времени использования мобильного устройства. В среднем, экранное время современного человека достигает 10-11 часов в день. Данные разнятся в зависимости от возраста. Рекомендуется пользоваться мобильным устройством от 2 до 4 часов в день. Чтобы сон был здоровым, необходимо исключить использование устройства за час-два до сна. При использовании мобильного телефона также необходимо четко отделять рабочее время от нерабочего. Это поможет дольше концентрироваться на необходимых процессах, не расходуя энергию на переключение с темы на тему;

– использование безопасных мобильных приложений. Киберпреступники овладели навыками создания и распространения вредоносных приложений, которые кажутся законными. При установке одного из таких приложений, преступники могут получить полный контроль над мобильным устройством или данными.

Для устройств Apple рекомендуется загружать только мобильные приложения из App Store. Преимущество здесь в том, что Apple выполняет проверку безопасности всех мобильных приложений, прежде чем они станут доступными для клиентов.

Для устройств Android рекомендуется загружать мобильные приложения только из Google Play, который поддерживается Google. Подобно Apple, Google проверяет безопасность всех приложений, прежде чем они станут доступными для клиентов.

Независимо от бренда необходимо изучить приложение перед его загрузкой: сколько времени было доступно мобильное приложение, сколько людей им пользовалось, и кто его продавец;

– защита персональных данных с помощью пароля или биометрии. Пароль предотвращает несанкционированный доступ к телефону, защищая личные данные, такие как фотографии, сообщения, контакты и другую конфиденциальную информацию. В случае кражи установленный пароль

					ПГУ1.090301.24.001 ПЗ	Лист
						77
Изм.	Лист	№ докум.	Подпись	Дата		

затруднит доступ к его содержимому для злоумышленника. Это может помочь сохранить личные данные в безопасности.

В целом, установка пароля на телефон является важным шагом для защиты личной жизни и конфиденциальной информации.

2.5 Электробезопасность при работе с ПК

При пользовании средствами вычислительной техники и периферийным оборудованием каждый работник должен внимательно и осторожно обращаться с электропроводкой, приборами и аппаратами.

Во избежание поражения электрическим током необходимо знать и выполнять следующие правила безопасного пользования электроэнергией:

- необходимо постоянно следить на своем рабочем месте за исправным состоянием электропроводки, выключателей, штепсельных розеток, при помощи которых оборудование включается в сеть, и заземления;
- при обнаружении неисправности немедленно обесточить электрооборудование, оповестить администрацию;
- продолжение работы возможно только после устранения неисправности.

Во избежание повреждения изоляции проводов и возникновения коротких замыканий не разрешается:

- вешать что-либо на провода;
- закрашивать и белить шнуры и провода;
- закладывать провода и шнуры за газовые и водопроводные трубы, за батареи отопительной системы;
- выдергивать штепсельную вилку из розетки за шнур, усилие должно быть приложено к корпусу вилки.

Для исключения поражения электрическим током запрещается:

- часто включать и выключать компьютер без необходимости;
- прикасаться к экрану и к тыльной стороне блоков компьютера;

- работать на средствах вычислительной техники и периферийном оборудовании мокрыми руками;
- работать на средствах вычислительной техники и периферийном оборудовании, имеющих нарушения целостности корпуса, нарушения изоляции проводов, неисправную индикацию включения питания с признаками электрического напряжения на корпусе;
- класть на средства вычислительной техники и периферийное оборудование посторонние предметы.

Запрещается под напряжением очищать от пыли и загрязнения электрооборудование.

Запрещается проверять работоспособность электрооборудования в непригодных для эксплуатации помещениях с токопроводящими полами, сырых, не позволяющих заземлить доступные металлические части.

Ремонт электроаппаратуры производится только специалистами-техниками с соблюдением необходимых технических требований.

Во всех случаях поражения человека электрическим током немедленно вызывают врача. До прибытия врача нужно, не теряя времени, приступить к оказанию первой помощи пострадавшему.

3 Стартап проект

Инициатор проекта планирует разработку кроссплатформенного мобильного приложения, которое будет предоставлять пользователям возможность записывать важную информацию, фиксировать ее в приложении, просматривать ее и эффективно ей управлять.

Целью деятельности предприятия является извлечение дохода для улучшения материального благосостояния его участника. Цель разрабатываемого продукта: упрощение рутинных задач, фиксирование важных аспектов рабочей или личной деятельности, повышение работоспособности пользователей, решение проблемы забывчивости. Необходимо предоставить пользователям современное решение, которое позволит им эффективно управлять своим временем, личными записями и организовывать свою деятельность.

Целевой аудиторией являются физические лица, люди, у которых есть потребность в сохранении информации и доступности к ней в любой момент времени.

На рынке присутствуют многочисленные аналоги, составляющие конкуренцию разрабатываемой системе. Среди них есть как малоизвестные приложения с небольшой аудиторией, так и приложения от крупных корпораций или более популярные в этом сегменте программные продукты.

Крупные приложения, составляющие конкуренцию: Microsoft OneNote, Evernote, Notion, Notex. Общее количество скачиваний приложений–конкурентов составляет 610 млн. и 20 тыс. скачиваний.

Перечисленные выше приложения ориентированы на мировой рынок и занимают 100% рынка. Разрабатываемое приложение на данном этапе поддерживает только русский язык и направлено на использование русскоязычной аудиторией. Приложение на начальном этапе неспособно привлечь в эту сферу новых и иноязычных пользователей, поэтому после

выпуска приложения в сторы количество пользователей составит не менее 5% от общего количества русскоязычной аудитории.

3.1 Сегменты ЦА

Целевая аудитория состоит из физических лиц разных возрастов и профессий. Для более эффективного взаимодействия с каждым сегментом ЦА, необходимо учитывать их потребности и предпочтения.

Целевая аудитория – люди, которые испытывают потребность в фиксации и организации информации. Приложение рассчитано на людей, которым необходимо записывать свои мысли, планы на день, итоги личных или деловых встреч. Также, целевой аудиторией могут быть люди, у которых есть потребность в постоянном доступе к электронной записной книжке, или те, которые предпочитают электронные записи бумажным вариантам – блокнотам, ежедневникам, тетрадям.

Целевая аудитория приложения «Заметки» довольно широкая, так как заметками пользуются люди разных возрастов, профессий и интересов.

Основные группы пользователей приложения «Заметки»:

- студенты и школьники. Они используют приложение для создания заметок по учёбе, конспектов, планов на будущее;
- бизнесмены и офисные работники. Для них приложение является незаменимым инструментом для ведения дел, планирования встреч, составления отчётов и других рабочих документов;
- фрилансеры и удалённые сотрудники. Эти люди часто работают дома или в общественных местах, поэтому им необходимо иметь под рукой удобный инструмент для создания заметок и напоминаний о важных делах;
- путешественники и туристы. Приложение помогает им сохранять информацию о бронировании билетов, гостиниц, экскурсий и других туристических услуг;

– люди, ведущие здоровый образ жизни. В приложении они могут создавать заметки о своём питании, тренировках и других аспектах здорового образа жизни.

Сегментация ЦА позволит определить каналы продвижения, способы более эффективного взаимодействия с аудиторией и сбор обратной связи.

Перед запуском проекта важно найти ключевых партнеров для ведения бизнеса, в них должны входить поставщики технического оборудования, квалифицированные кадры, инвесторы.

Поставщиками технического оборудования могут стать компании, которые предоставляют компьютеры, серверы и другое необходимое оборудование для функционирования разрабатываемой системы.

Для дальнейшей разработки, развития, улучшения и продвижения приложения требуются квалифицированные кадры: программисты, дизайнеры, архитекторы, SMM–специалисты.

Инвесторами могут стать лица, заинтересованные в развитии данного программного обеспечения.

3.2 Ключевые процессы

Ниже перечислены основные ключевые процессы при разработке стартап–проекта:

- исследование рынка;
- создание бизнес–плана;
- поиск средств;
- поиск и создание команды;
- разработка ПО;
- продвижение;
- поддержка продукта;
- мониторинг и анализ.

Исследование рынка: прежде чем начать разработку проекта, важно провести исследование рынка, чтобы понять потребности и предпочтения

					ПГУ1.090301.24.001 ПЗ	Лист
						82
Изм.	Лист	№ докум.	Подпись	Дата		

потенциальных клиентов. Это поможет определить, есть ли спрос на предлагаемый продукт. Исследование потребностей ЦА и особенностей рынка представлено в п. 3 и 3.1.

Создание бизнес–плана: после проведения исследования рынка следующим шагом будет создание бизнес–плана. Он должен содержать детальное описание продукта или услуги, анализ конкурентов, маркетинговую стратегию, финансовые прогнозы и другие важные аспекты бизнеса.

Поиск средств: после разработки бизнес–плана необходимо определить основные источники финансирования. Источниками могут стать: собственные средства и сбережения, государственная поддержка – конкурсы, форумы, проекты с призовым фондом, грантом или льготным кредитом. А также некоторые банки предлагают специальные программы для финансирования стартапов, имеющих бизнес–план.

Построение команды: для успешной разработки стартап–проекта необходима сильная команда с разными навыками и опытом работы. Важно найти людей, которые разделяют видение проекта и готовы работать над его реализацией.

Этапы разработки ПО описаны в пункте 1.

Продвижение: после того, как продукт или услуга были разработаны и протестированы, следующим шагом будет продвижение на рынке. Продвижение может включать в себя маркетинговые кампании, партнерства с другими компаниями и участие в мероприятиях для привлечения внимания к проекту.

Поддержка продукта: в процессе разработки необходимо вносить улучшения, устранять дефекты программного обеспечения и оптимизировать работу приложения после передачи его в эксплуатацию. В задачи технической поддержки приложения входит выявление недоработок, багов, сбор обратной связи от пользователей с целью передачи этих сведений команде разработки. В ходе поддержки продукта выявленные ошибки

					<i>ПГУ1.090301.24.001 ПЗ</i>	Лист
						83
Изм.	Лист	№ докум.	Подпись	Дата		

устраняются, а также вносятся изменения по добавлению нового функционала, повышающие удобство использования приложения. Техническая поддержка предоставляет помощь пользователям в решении конкретных проблем с продуктом и его использованием.

Мониторинг и анализ: наконец, важно следить за результатами проекта и анализировать отзывы пользователей, чтобы понимать, какие аспекты работают стабильно, а какие нуждаются в улучшении. Это поможет сделать необходимые корректировки и продолжать развивать стартап–проект.

3.3 Каналы взаимодействия

Каналы взаимодействия с целевой аудиторией стартап проекта – это способы и средства, которые позволяют донести информацию о проекте до потенциальных клиентов и партнеров.

Для взаимодействия с целевой аудиторией стоит рассмотреть следующие каналы:

- сайт. Создание сайта позволит отобразить всю необходимую информацию о проекте. На сайте будет представлен функционал, который предлагает информационная система, контактная информация, тарифные планы и другие важные детали. Сайт предоставляет потенциальным пользователям доступ к информации о компании и проекте в любое время. А также позволит взаимодействовать с большей частью целевой аудитории;

- социальные сети. Создание страницы проекта в популярных социальных сетях, таких как ВКонтакте, Telegram, Discord, TikTok, X позволит взаимодействовать с молодой частью целевой аудитории: школьниками и студентами. Публикация новостей о проекте, проведение конкурсов и акций позволит повысить интерес к продукту и привлечет новых пользователей;

- блоги и форумы. Активность в тематических блогах и на форумах позволит охватить часть целевой аудитории, увлекающуюся туризмом и путешествиями, а также часть аудитории, придерживающуюся ЗОЖ.

Комментарии под постами и ответы на вопросы пользователей помогут аудитории узнать больше о разрабатываемом продукте;

- реклама. Использование контекстной рекламы в поисковых системах Google AdWords или Яндекс.Директ позволит привлечь фрилансеров и офисных работников, так как эта часть целевой аудитории тесно взаимодействует с поисковыми системами. Также можно размещать баннеры на сайтах с высокой посещаемостью;

- партнерские программы. Заключение партнерских соглашений с другими компаниями позволит продвигать проект через их каналы коммуникации. Примером акции, направленной на расширение аудитории продукта, может служить предоставление скидки на свои услуги для клиентов партнера.

3.4 Источники дохода

Источники дохода для стартапа зависят от конкретной бизнес–модели и отрасли, в которой планируется развертывание проекта.

Проект будет иметь следующие источники дохода:

- подписка или абонентская плата. Приложение может иметь несколько тарифов или вариантов подписок/абонентских плат за использование расширенного функционала. Это может быть ежемесячная или ежегодная оплата. Стоимость ежемесячной и ежегодной оплаты будет составлять примерно 200 рублей и 1700 рублей соответственно;

- реклама. Еще один источник заработка – размещение рекламы в приложении. Доход может зависеть от кликов по рекламным объявлениям или от продажи рекламных мест. Реклама может быть отключена за одноразовую оплату в размере 100 рублей;

- инвестиции. Необходимо будет привлечь инвесторов в обмен на долю в компании. Этот источник дохода может быть временным, пока проект не станет прибыльным.

Доход высчитывается по формуле:

					ПГУ1.090301.24.001 ПЗ	Лист
						85
Изм.	Лист	№ докум.	Подпись	Дата		

$$Д = К * Ц, \quad (1)$$

где Д – доход;

К – количество потребителей;

Ц – установленная цена.

Доход от продажи ежемесячных подписок высчитывается:

$$400\text{п} * 200\text{р} = 80\,000\text{р/мес.}$$

Доход от продажи ежегодной подписки высчитывается:

$$300\text{п} * 1700\text{р} = 510\,000\text{р/год.}$$

Доход от ежемесячной продажи функции «Отключение рекламы» высчитывается:

$$300\text{п} * 100\text{р} = 30\,000\text{р/мес.}$$

Таким образом, общий ожидаемый доход составит 1 830 000 рублей ежегодно и 152 500 рублей ежемесячно.

3.5 Ключевые ресурсы

Для запуска, функционирования и развития бизнеса потребуются некоторые ключевые ресурсы. В нашем случае это будут:

- разработчики программного обеспечения, специализирующиеся на создании информационных систем;
- дизайнеры, специализирующиеся на подготовке макетов и отрисовке дизайна приложения;
- SMM–специалисты, занимающиеся продвижением продукта через социальные медиа платформы;
- компьютеры, серверы, программное обеспечение и другое техническое оборудование для разработки и обслуживания системы;
- офисное пространство для работы команды;
- ресурсы для финансирования разработки, маркетинга, продаж и операционных расходов бизнеса.

Финансирование проекта планируется осуществить как за счет собственных средств инициатора проекта, так и за счет государственной

					ПГУ1.090301.24.001 ПЗ	Лист
						86
Изм.	Лист	№ докум.	Подпись	Дата		

поддержки «Гранты для молодых предпринимателей». Ниже в таблице 16 можно увидеть программу финансирования.

Таблица 16 – Источники финансирования

Источник финансирования	Сумма
Собственные средства	500 000р
Гос. поддержка	500 000р
Всего	1 000 000р

Приведенные выше ключевые ресурсы можно разделить на финансовые и людские. Направление использования средств подробно указано в таблицах 17 и 18.

Таблица 17 – Финансовые ключевые ресурсы

Направление использования средств	Кол-во, шт	Цена, руб.	Сумма, руб
Аренда помещения (6мес.)	6	4000	24 000
Закупка ноутбуков	2	100 000	200 000
Закупка другого оборудования	1	50 000	50 000
Ком. услуги	6	5 000	30 000

Таблица 18 – Людские ключевые ресурсы

Направление использования средств	Кол-во, шт	Цена, руб.	Сумма, руб
З/п разработчику (6 мес)	6	50 000	300 000
З/п дизайнеру (6 мес)	6	30 000	180 000
З/п бухгалтерской компании (6 мес)	6	10 000	60 000
З/п SMM-специалисту (6 мес)	6	30 000	180 000

3.6 Риски проекта

Реализация стартап–проекта может быть подвержена различным рискам и форс–мажорным обстоятельствам, которые могут оказать негативное влияние на успех проекта.

Возможными рисками являются, например:

- рыночный риск. Если стартап предлагает продукт, который не соответствует потребностям рынка или не решает реальную проблему, он может столкнуться с трудностями в привлечении клиентов и получении прибыли;
- конкурентный риск. На рынке существует жесткая конкуренция, и, если стартап не может предложить уникальное решение или преимущество перед своими конкурентами, то возможна потеря доли рынка и прибыли;
- финансовый риск. Запуск стартапа требует значительных инвестиций, и, если проект не приносит ожидаемых доходов, он может столкнуться с финансовыми проблемами;
- технологический риск. Стартап, работающий в сфере технологий может столкнуться с техническими проблемами, такими как сбои в работе программного обеспечения или проблемы совместимости с другими системами;
- риск изменения обстоятельств. Внешние факторы, такие как экономические кризисы, политические события или технологические инновации, могут оказать негативное влияние на стартап и его бизнес–модель.

Важно провести анализ и предусмотреть меры по минимизации возможных негативных последствий таких обстоятельств.

Существует несколько способов снижения рисков при разработке программного обеспечения:

- тщательное исследование рынка и потребностей пользователей. Исследование может включать опросы, интервью, фокус-группы и бета-

тестирование. Исследование рынка поможет лучше понять размер рынка, спрос, предпочтения и конкурентную среду. Данный способ снижения рисков может помочь адаптировать свой продукт и цены на него в соответствии с потребностями и ожиданиями рынка;

- на этапе планирования проекта необходимо провести тщательный анализ функциональных и бизнес-требований, чтобы убедиться, что они полностью понятны и реалистичны. При планировании проекта также важным шагом является составление детального плана работ, который поможет избежать проблем с временем выполнения задач и оптимизировать процесс разработки;

- подбор квалифицированных сотрудников. Профессиональная команда разработки может значительно повысить качество продукта, сократить время его создания, эффективно решать сложные задачи и принимать правильные технические решения. Они знают какие инструменты использовать, как оптимизировать код и как избежать распространенных ошибок. Такой подход поможет избежать сбоев в работе программного продукта, предотвратить возникновение багов и обеспечить безопасность и надежность разрабатываемого продукта.

3.7 Бизнес–план

На первом этапе приложение представляет собой минимально жизнеспособный продукт (MVP) – продукт или версия, в которой уже есть функции, удовлетворяющие потребности пользователей и которая служит для демонстрации возможностей ПО клиенту.

Основная задача на данном этапе – получение обратной связи для формирования гипотез дальнейшего развития продукта. Сбор информации от MVP зачастую дешевле, чем разработка продукта с большим количеством функций.

На этапе MVP в команду разработки будут входить: программист, SMM–специалист и бухгалтер. Съем офисного помещения не требуется, так

					ПГУ1.090301.24.001 ПЗ	Лист
						89
Изм.	Лист	№ докум.	Подпись	Дата		

как сотрудники будут работать в дистанционном формате. Из технического оборудования потребуются ноутбуки и мобильные устройства на ОС Android и iOS. Основным каналом для продвижения будут социальные сети и рекламные объявления. На данном этапе сумма вложений будет составлять около 200 000 рублей.

В процессе пилотирования проекта собирается обратная связь, отзывы пользователей, обращения в службу поддержки и комментарии в социальных сетях. По итогам запуска приложения необходимо проанализировать отзывы пользователей и сделать выводы об успешности продукта и перспективах, а также устранить недостатки, сформировать задачи для дальнейшего процесса разработки.

В случае успеха MVP разработка проекта будет требовать больше кадровых и финансовых ресурсов. Команда разработки будет нуждаться в архитекторе, дизайнере и тестировщике. С масштабированием команды разработки возникнет необходимость в съеме офисного помещения, закупке дополнительного технического оборудования для удовлетворения требований команды разработки. Сумма дополнительных вложений будет составлять 800 000 рублей.

Для запуска проекта необходим стартовый капитал в размере 1 000 000 рублей. По предварительным расчетам общий ожидаемый доход составит 1 830 000 рублей в год. В случае успеха проект окупится в течение полугода.

Таким образом, можно говорить об успешности проекта и целесообразности разработки MVP.

					ПГУ1.090301.24.001 ПЗ	Лист
						90
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

В ходе выполнения выпускной квалификационной работы было разработано кроссплатформенное мобильное приложение «Заметки», обладающее удобным и интуитивно понятным пользовательским интерфейсом.

Данный проект представляет собой приложение, позволяющее записывать важную информацию, список дел, мысли и итоги деловых или личных встреч. Приложение позволяет просматривать зафиксированную информацию, а также эффективно управлять ей. В результате разработки приложение «Заметки» является кроссплатформенным: функционирует на смартфонах на платформах iOS и Android.

Разработка данного приложения позволила улучшить навыки в работе с базами данных и навыки создания клиентских приложений сразу для двух платформ: iOS и Android.

В процессе реализации данного проекта были достигнуты все поставленные цели. Проект предоставляет все необходимые функции для работы с личными записями и соответствует заявленным требованиям. Проведенное тестирование и отладка показали, что приложение «Заметки» работает исправно и выполняет все необходимые задачи.

Перспективы развития системы включают расширение функционала, добавление новых функций, улучшение пользовательского интерфейса и оптимизацию процессов.

Разработанная информационная система готова к использованию в личных или рабочих целях и может значительно повысить работоспособность, поможет расставить приоритеты и отслеживать текущие задачи.

Список использованных источников

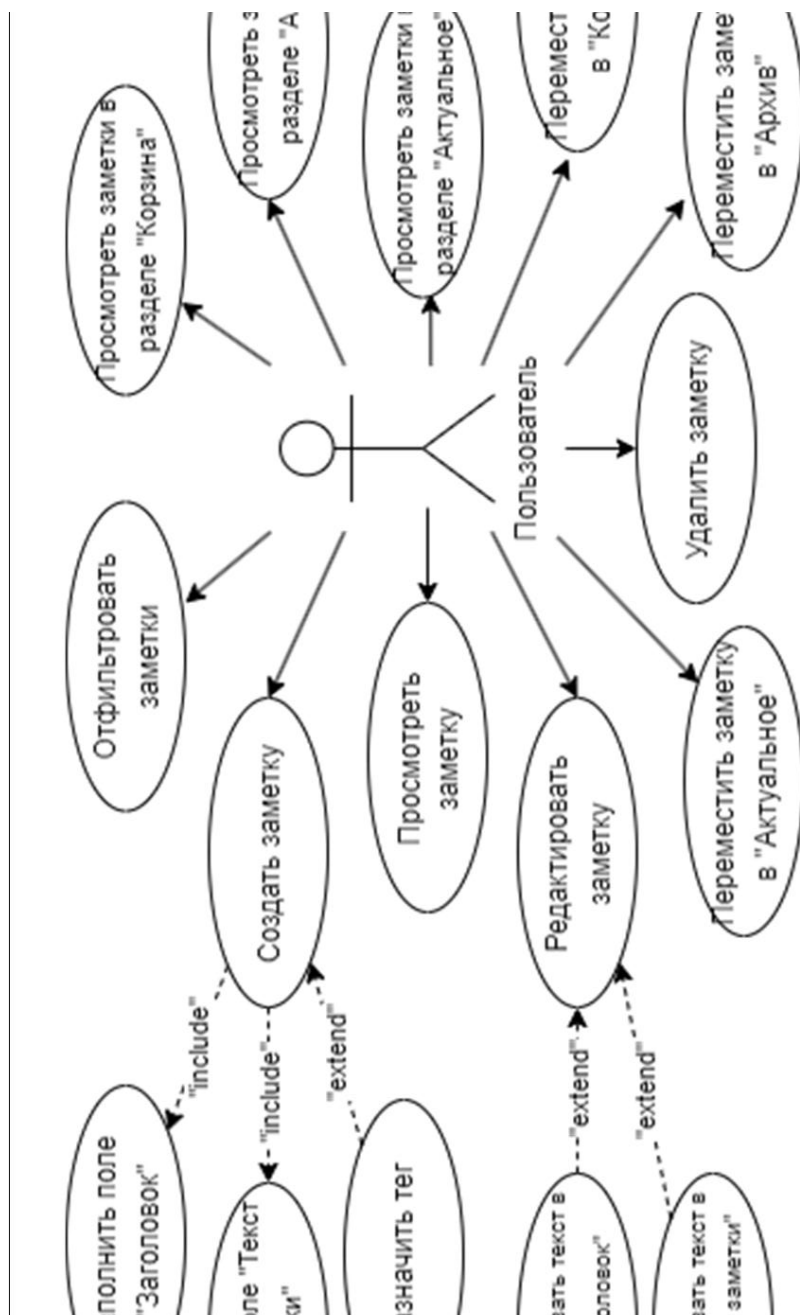
1. Чернышев С.А. Основы Dart, 2-е издание, переработанное и дополненное, 2024.
2. Андрей Алеев. Быстрый старт Flutter-разработчика. – Екб.: «Издательские решения», 2020.
3. Flutter – Dart API docs [Электронный ресурс] URL: <https://main-api.flutter.dev> (дата обращения 01.05.2024)
4. SQLite Documentation [Электронный ресурс] URL: <https://www.sqlite.org/docs.html> (дата обращения 02.05.2024)
5. Требования к рабочему месту программиста – Мероприятия по охране труда инженера-программиста [Электронный ресурс] URL: <https://subject-book.com/oxrana-truda/trebovaniya-k-rabochemu-mestu-programmista-meropriyatiya-po-oxrane-truda-inzhenera-programmista.html> (дата обращения 18.05.2024)
6. Основные производственные опасные и вредные факторы, воздействующие на программиста [Электронный ресурс] URL: https://studbooks.net/1385927/bzhd/osnovnye_proizvodstvennye_opasnye_i_vrednye_faktory_vozdeystvuyuschie_na_programmista (дата обращения 19.05.2024)
7. Гид по максимально безопасному использованию смартфона для детей и взрослых [Электронный ресурс] URL: <https://beeline.ru/articles/gid-po-maksimalno-bezopasnomu-ispolzovaniyu-smartfona-dlya-detey-i-vzroslykh/> (дата обращения 20.05.2024)
8. Оценка и снижение маркетинговых рисков: стартапы и управление маркетинговыми рисками: стратегии успеха [Электронный ресурс] URL: <https://fastercapital.com/ru/content/Оценка-и-снижение-маркетинговых-рисков-стартапы-и-управление-маркетинговыми-рисками-стратегии-успеха.html> (дата обращения 24.05.2024)

Приложение А

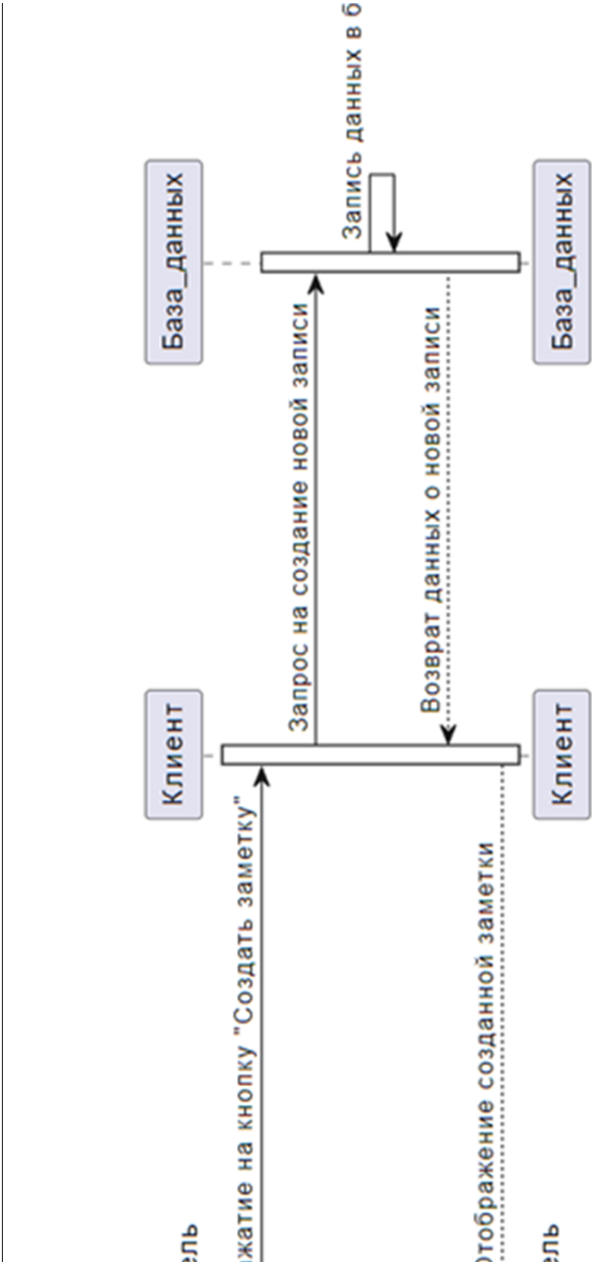
(Обязательное)

Стартап-проект. Кроссплатформенное мобильное приложение «Заметки»
UML-диаграммы

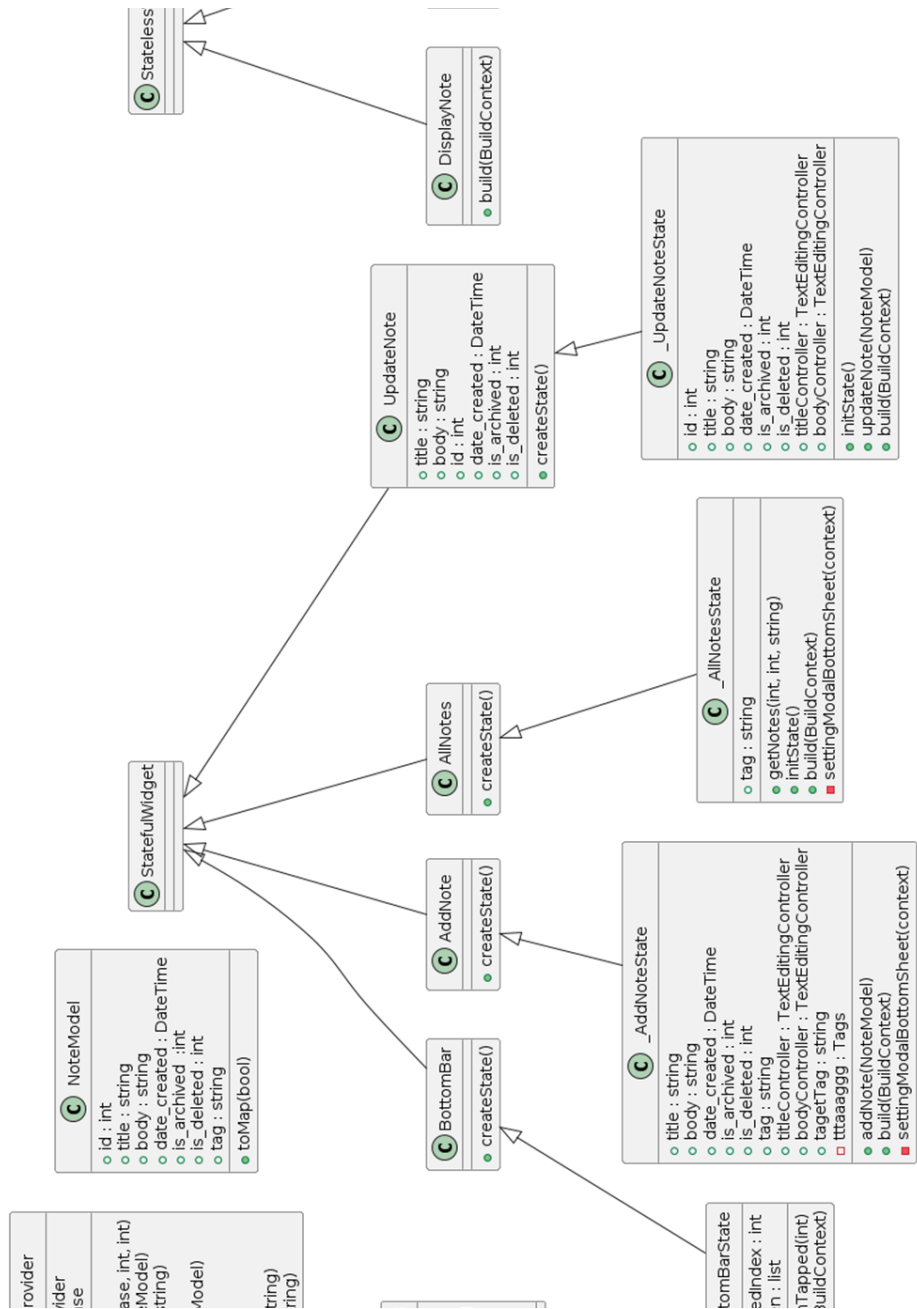
Приложение А.1 – Диаграмма прецедентов



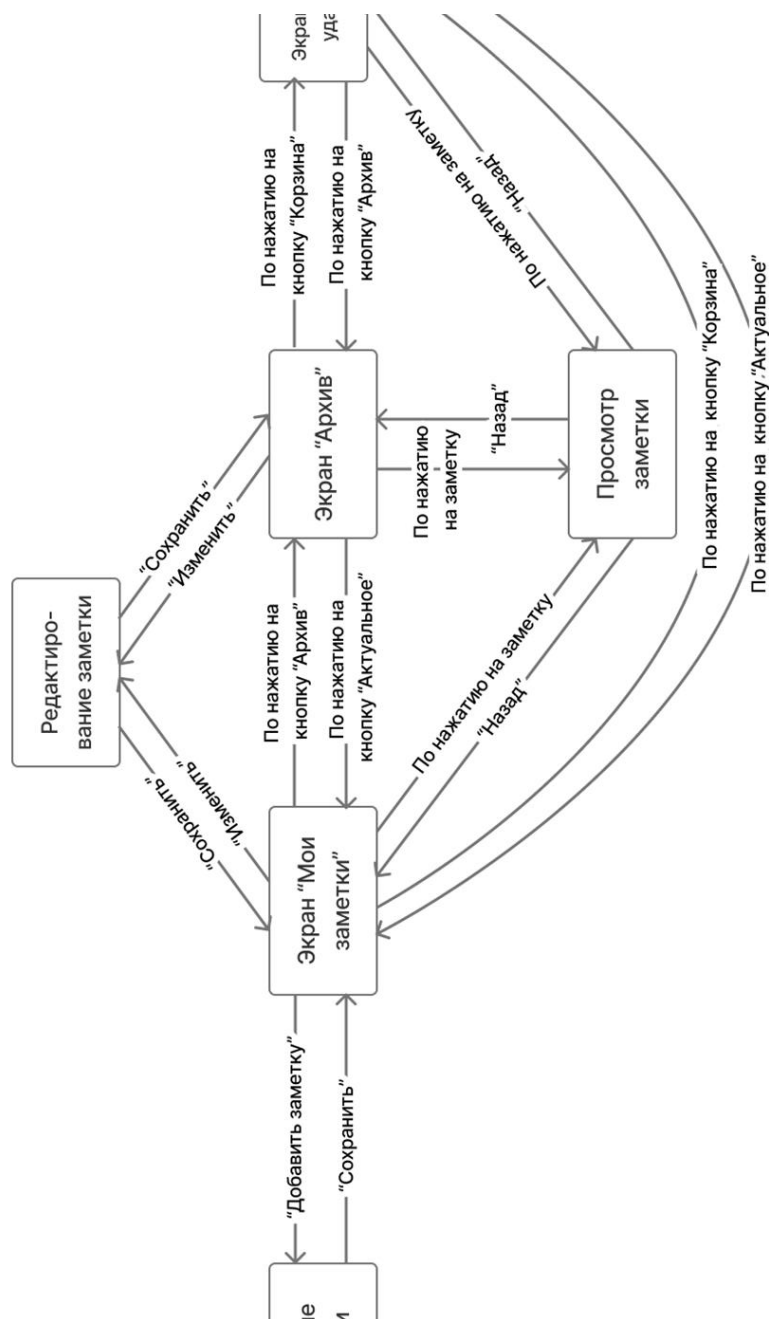
Приложение А.2 – Диаграмма последовательности «Создание новой заметки»



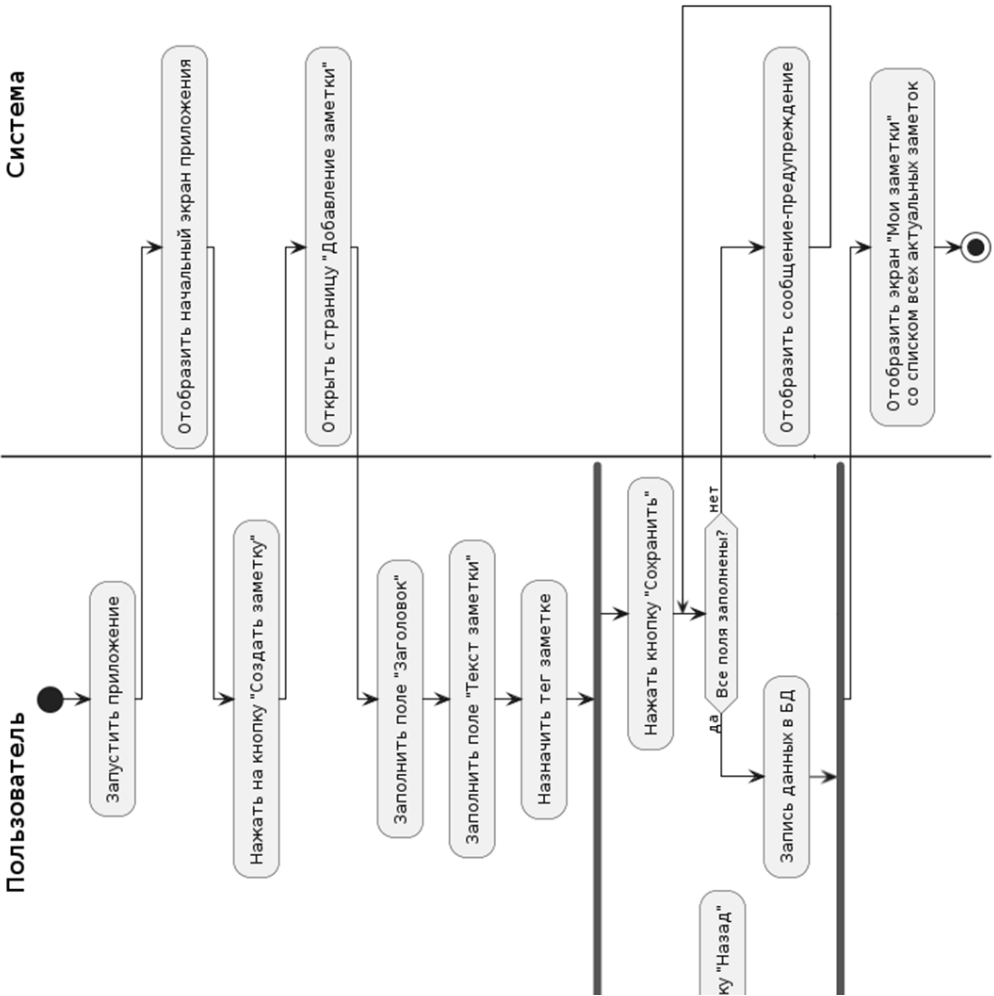
Приложение А.3 – Диаграмма классов



Приложение А.4 – Карта экранов



Приложение А.5 – Диаграмма деятельности «Создание заметки»



Приложение Б

(Обязательное)

Стартап-проект. Кроссплатформенное мобильное приложение «Заметки»
Листинг программы

Файл database_provider.dart

```
import 'package:sqflite/sqflite.dart';
import 'dart:async';
import 'package:path/path.dart';
import '../model/note_model.dart';

class DatabaseProvider {
  DatabaseProvider._();

  static final DatabaseProvider db = DatabaseProvider._();
  static Database? _database;

  Future <Database?> get database async {
    if (_database != null) {
      return _database;
    }
    _database = await initDB();
    return _database;
  }

  initDB() async {
    return await openDatabase(join(await getDatabasesPath(),
"notes_app.db"),
      onCreate: (db, version) async {
        await db.execute('''
CREATE TABLE all_notes(
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  title BLOB,
  body BLOB,
  date_created DATE,
  is_archived INT,
  is_deleted INT,
  tag STRING
)''')
      },
    );
  }
}
```

					ПГУ1.090301.24.001 ПЗ	Лист
						100
Изм.	Лист	№ докум.	Подпись	Дата		

```

        );
    }, version: 4);
}

// UPGRADE DATABASE TABLES
/* void _onUpgrade(Database db, int oldVersion, int newVersion)
{
    if (oldVersion < newVersion) {
        db.execute("ALTER TABLE all_notes ADD COLUMN tag
STRING;");
    }
}*/

addNewNote(NoteModel note) async {
    final db = await database;
    db?.insert("all_notes", note.toMap(false),
        conflictAlgorithm: ConflictAlgorithm.replace);
}

Future<dynamic> getNotes(is_archived, is_deleted, String? tag)
async {
    final db = await database;

    var res = tag == null? await db?.rawQuery("SELECT * FROM
all_notes WHERE is_archived = ? and is_deleted = ?",
        [is_archived=0, is_deleted = 0]):
        await db?.rawQuery("SELECT * FROM all_notes WHERE
is_archived = ? and is_deleted = ? and tag = ?",
            [is_archived=0, is_deleted = 0, tag]);
    if (res!.isEmpty) {
        return null;
    } else {
        var resultMap = res.toList();
        return resultMap.isNotEmpty ? resultMap : null;
    }
}

```

```

Future<int?> deleteNote(int id) async{
    final db = await database;

    int? count = await db?.rawDelete("DELETE FROM all_notes
WHERE id = ?", [id]);

    return count;
}

updateNote(NoteModel note) async {
    final db = await database;

    db?.rawUpdate('UPDATE all_notes SET title = ?, body = ?,
date_created = ?, is_archived = ?, is_deleted = ? WHERE id = ?',
[note.title, note.body, note.date_created.toString(),
note.is_archived, note.is_deleted, note.id]);
}

toArchive (int id) async {
    final db = await database;

    db?.rawUpdate('UPDATE all_notes SET is_archived = 1,
is_deleted = 0 WHERE id = ?', [id]);
}

toActual (int id) async {
    final db = await database;

    db?.rawUpdate('UPDATE all_notes SET is_archived = 0,
is_deleted = 0 WHERE id = ?', [id]);
}

toDelete (int id) async {
    final db = await database;

    db?.rawUpdate('UPDATE all_notes SET is_archived = 0,
is_deleted = 1 WHERE id = ?', [id]);
}

Future<dynamic> getArchived (is_archived, String? tag) async
{
    final db = await database;

```

					ПГУ1.090301.24.001 ПЗ	Лист
						102
Изм.	Лист	№ докум.	Подпись	Дата		

var res = tag == null? await db?.rawQuery("SELECT * FROM

```

all_notes WHERE is_archived = ?", [is_archived=1]):

    await db?.rawQuery("SELECT * FROM all_notes WHERE
is_archived = ? and tag = ?",
        [is_archived=1, tag]);
    if (res!.isEmpty) {
        return null;
    } else {
        var resultMap = res.toList();
        return resultMap.isNotEmpty ? resultMap : null;
    }
}

Future<dynamic> getDeleted (is_deleted, String? tag) async {
    final db = await database;

    var res = tag == null? await db?.rawQuery("SELECT * FROM
all_notes WHERE is_deleted = ?", [is_deleted=1]):

    await db?.rawQuery("SELECT * FROM all_notes WHERE is_deleted
= ? and tag = ?",
        [is_deleted=1, tag]);
    if (res!.isEmpty) {
        return null;
    } else {
        var resultMap = res.toList();
        return resultMap.isNotEmpty ? resultMap : null;
    }
}
}

```

Файл note_model.dart

```

class NoteModel {
    int? id;
    String? title;
    String? body;
    DateTime? date_created;
    int? is_archived = 0;
}

```

					ПГУ1.090301.24.001 ПЗ	Лист
						104
Изм.	Лист	№ докум.	Подпись	Дата		

```

int? is_deleted = 0;
String? tag;

NoteModel({this.id, this.title, this.body, this.date_created,
this.is_archived, this.is_deleted, this.tag});

Map<String, dynamic> toMap(bool forUpdate) {
  var data = {
    'id': id,
    'title': title,
    'body': body,
    'date_created': date_created.toString(),
    'is_archived' : is_archived,
    'is_deleted' : is_deleted,
    'tag' : tag,
  };
  if(forUpdate){
    data['id'] = this.id;
  }
  return data;
}
}

```

Файл note_tags.dart

```

import 'dart:ui';
import 'package:flutter/material.dart';

Color getColor(String tag){
  switch (tag)
  {
    case 'Работа':
      return Color(0xffff1493).withOpacity(0.9);
    case 'Учеба':
      return Color(0xff002aff).withOpacity(0.8);
    case 'Личное' :

```

					ПГУ1.090301.24.001 ПЗ	Лист
						105
Изм.	Лист	№ докум.	Подпись	Дата		

```

        return Color(0xff00cf07).withOpacity(0.8);
    case 'Задачи' :
        return Color(0xffff4d00).withOpacity(0.8);
    case 'Финансы' :
        return Color(0xffff4a900).withOpacity(0.9);
    }
    return Colors.white.withOpacity(0.9);
}

```

Файл add_note.dart

```

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:new_notes/database/database_provider.dart';
import '../model/note_model.dart';

```

```

class AddNote extends StatefulWidget {
  AddNote({Key? key}) : super(key: key);

  @override
  _AddNoteState createState() => _AddNoteState();
}

```

```

enum Tags { work, study, personal, tasks, money, none }

```

```

class _AddNoteState extends State<AddNote> {
  String? title;
  String? body;
  DateTime? date_created;
  int? is_archived;
  int? is_deleted;
  String? tag;

```

```

    TextEditingController titleController =
    TextEditingController();

```

					ПГУ1.090301.24.001 ПЗ	Лист
						106
Изм.	Лист	№ докум.	Подпись	Дата		

```

        TextEditingController          bodyController          =
TextEditingController();

        String tagetTag = '';
        Tags? _tttaaaggg;

        addNote(NoteModel note) {
            DatabaseProvider.db.addNewNote(note);
            print("заметка добавлена в бд");
        }

        @override
        Widget build(BuildContext context) {
            return Scaffold(
                appBar: AppBar(
                    title: Text("Добавление заметки"),
                    backgroundColor: const Color(0xffffcc0dd),
                    actions: [
                        IconButton(
                            onPressed: () =>
                                _settingModalBottomSheet(context),
                            icon: Icon(Icons.new_label))
                    ],
                ),
                body: Padding(
                    padding: const EdgeInsets.symmetric(vertical: 8.0,
horizontal: 12.0),
                    child: Column(
                        children: [
                            TextField(
                                controller: titleController,
                                decoration: InputDecoration(
                                    border: InputBorder.none,
                                    hintText: "Заголовок",
                                ),
                            ),

```

```

        style: TextStyle(fontSize: 28.0, fontWeight:
FontWeight.bold),
      ),
      Expanded(
        child: TextField(
          controller: bodyController,
          keyboardType: TextInputType.multiline,
          maxLines: null,
          decoration: InputDecoration(
            border: InputBorder.none,
            hintText: "Текст заметки",
          ),
          style: TextStyle(fontSize: 20.0),
        ),
      ),
    ],
  )),
  floatingActionButton: FloatingActionButton.extended(
    onPressed: () {
      if (titleController.text == '' || bodyController.text
== '') {
        showDialog(
          context: context,
          builder: (ctx) => AlertDialog(
            title: const Text(
              "Необходимо заполнить все поля!",
              style: const TextStyle(
                fontSize: 24.0,
                fontWeight:
FontWeight.bold),
            ),
            content: const Text(
              "Заполните поля 'Заголовок' и 'Текст
заметки'",
              style: const TextStyle(
                fontSize: 18.0,

```

```

        ),
    ),
    actions: <Widget>[
        TextButton(
            onPressed: () {
                Navigator.of(context).pop();
            },
            child: Container(
                //padding: const EdgeInsets.all(14),
                child: const Text(
                    "Ок",
                    style: const TextStyle(
                        color: const Color(0xfffc4ca1),
                        fontSize: 18.0,
                    ),
                ),
            ),
        ),
    ],
),
);
} else {
    setState(() {
        title = titleController.text;
        body = bodyController.text;
        date_created = DateTime.now();
        is_archived = 0;
        is_deleted = 0;
        tag = tagetTag;
    });
    NoteModel note = NoteModel(
        title: title,
        body: body,
        date_created: date_created,

```

					ПГУ1.090301.24.001 ПЗ	Лист
						109
Изм.	Лист	№ докум.	Подпись	Дата		

is_archived: 0,
is_deleted: 0,

					ПГУ1.090301.24.001 ПЗ	Лист
						110
Изм.	Лист	№ докум.	Подпись	Дата		

```

        tag: tagetTag);

        addNote(note);

        Navigator.pushNamedAndRemoveUntil(context, "/",
(route) => false);
    }
},
label: Text("Сохранить"),
icon: Icon(Icons.save),
backgroundColor: const Color(0xffffcc0dd),
),
);
}

```

```

void _settingModalBottomSheet(context) {
    showModalBottomSheet(
        context: context,
        builder: (BuildContext bc) {
            return StatefulBuilder(
                builder: (BuildContext context, StateSetter
setState) {
                    return SizedBox(
                        child: Column(children: [
                            Padding(
                                padding: EdgeInsets.all(16.0),
                                child: Text(
                                    "Выберите тег",
                                    style: const TextStyle(
                                        fontSize: 20.0,
                                        fontWeight: FontWeight.bold,
                                    ),
                                ),
                            ),
                            Column(
                                children: <Widget>[

```

```

RadioListTile(
  title: const Text(
    'Работа',
    style: const TextStyle(
      fontSize: 18.0,
    ),
  ),
  value: Tags.work,
  groupValue: _tttaaaggg,
  onChanged: (Tags? value) {
    setState(() {
      _tttaaaggg = value;
      tagetTag = 'Работа';
      Navigator.of(context).pop();
    });
  },
),
RadioListTile(
  title: const Text(
    'Учеба',
    style: const TextStyle(
      fontSize: 18.0,
    ),
  ),
  value: Tags.study,
  groupValue: _tttaaaggg,
  onChanged: (Tags? value) {
    setState(() {
      _tttaaaggg = value;
      tagetTag = 'Учеба';
      Navigator.of(context).pop();
    });
  },
),

```

```

),
RadioListTile(
  title: const Text(
    'Личное',
    style: const TextStyle(
      fontSize: 18.0,
    ),
  ),
  value: Tags.personal,
  groupValue: _tttaaaggg,
  onChanged: (Tags? value) {
    setState(() {
      _tttaaaggg = value;
      tagetTag = 'Личное';
      Navigator.of(context).pop();
    });
  },
),
RadioListTile(
  title: const Text(
    'Задачи',
    style: const TextStyle(
      fontSize: 18.0,
    ),
  ),
  value: Tags.tasks,
  groupValue: _tttaaaggg,
  onChanged: (Tags? value) {
    setState(() {
      _tttaaaggg = value;
      tagetTag = 'Задачи';
      Navigator.of(context).pop();
    });
  },
),

```

```

    },
  ),
  RadioListTile(
    title: const Text(
      'Финансы',
      style: const TextStyle(
        fontSize: 18.0,
      ),
    ),
    value: Tags.money,
    groupValue: _tttaaaggg,
    onChanged: (Tags? value) {
      setState(() {
        _tttaaaggg = value;
        tagetTag = 'Финансы';
        Navigator.of(context).pop();
      });
    },
  ),
  RadioListTile(
    title: const Text(
      'Без тега',
      style: const TextStyle(
        fontSize: 18.0,
      ),
    ),
    value: Tags.none,
    groupValue: _tttaaaggg,
    onChanged: (Tags? value) {
      setState(() {
        _tttaaaggg = value;
        tagetTag = "";
        Navigator.of(context).pop();
      });
    },
  ),

```

```

        });
    },
    ),
],
)
]),
);
},
);
});
}
}

```

Файл all_notes.dart

```

import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:new_notes/database/database_provider.dart';
import 'package:new_notes/screens/update_note.dart';
import '../model/note_model.dart';
import 'package:intl/intl.dart';
import '../model/note_tags.dart';

class AllNotes extends StatefulWidget {
  @override
  _AllNotesState createState() => _AllNotesState();
}

class _AllNotesState extends State<AllNotes> {
  String? tag = null;

  getNotes(is_archived, is_deleted, String? tag) async {
    return await DatabaseProvider.db.getNotes(is_archived,
is_deleted, tag);
  }
}

```

```

@override
void initState() {
    super.initState();
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: const Color(0xffff0e1e8),
        appBar: AppBar(
            title: Text("Мои заметки"),
            backgroundColor: const Color(0xffffcc0dd),
            actions: [
                IconButton(
                    icon: Icon(Icons.filter_alt),
                    onPressed: () {
                        _settingModalBottomSheet(context);
                    },
                )
            ],
        ),
        body: FutureBuilder(
            future: getNotes(0, 0, tag),
            builder: (context, AsyncSnapshot<dynamic> noteData) {
                switch (noteData.connectionState) {
                    case ConnectionState.waiting:
                        {
                            return Center(child:
CircularProgressIndicator());
                        }
                    case ConnectionState.done:
                        {
                            if (noteData.data == null) {
                                print("пока пусто");
                                return Center(

```

```

        child: Text("Вы пока не создали ни одной
заметки"),

    );
} else {
    print("отображение содержимого бд");
    return Padding(
        padding: const EdgeInsets.all(8.0),
        child: ListView.builder(
            itemCount: noteData.data.length,
            itemBuilder: (context, index) {
                int id = noteData.data[index]['id'];
                String title =
noteData.data[index]['title'];
                String body =
noteData.data[index]['body'];
                String date_created =
noteData.data[index]['date_created'];
                int is_archived =
noteData.data[index]['is_archived'];
                int is_deleted =
noteData.data[index]['is_deleted'];
                String tag =
noteData.data[index]['tag'];
                return Card(
                    child: Column(children: [
                        SizedBox(
                            height: 8,
                        ),
                        ListTile(
                            title: Row(
                                children: [
                                    Text(
                                        title,
                                        maxLines: 1,
                                        overflow:

```

```

TextOverflow.ellipsis,

style: TextStyle(
    fontWeight:
FontWeight.bold),

),
Spacer(),
tag.isNotEmpty
? Container(
    width: 80,
    //height: 40,
    padding:

EdgeInsets.all(2),

    decoration:

BoxDecoration(

        color:

        borderRadius:

            topRight:

            topLeft:

            bottomLeft:

            bottomRight:

        ),
    child: Center(
        child: Text(tag,
            textAlign:

            style:      const

            color:

Colors.white,

        )),

```

					ПГУ1.090301.24.001 ПЗ	Лист
						118
Изм.	Лист	№ докум.	Подпись	Дата		

```

        ),
    ),
    : SizedBox.shrink()
  ],
),
subtitle: Text(
  body,
  maxLines: 3,
  overflow:
TextOverflow.ellipsis,
),
onTap: () {
  //NoteModel      note      =
noteData.data[index] as NoteModel;
  Navigator.pushNamed(context,
    "/DisplayNote",
    arguments: NoteModel(
      id: id,
      title: title,
      body: body,
      date_created:
DateTime.parse(date_created),
      is_archived:
is_archived,
      is_deleted: is_deleted,
    ));
},
),
Row(
  children: [
    SizedBox(
      width: 15,
    ),
    IconButton(

```

```

        icon: Icon(Icons.edit),
        onPressed: () {
            Navigator.push(
                context,
                MaterialPageRoute(
                    builder: (context)
=> UpdateNote(
                                id: id,
                                title:
                                body: body,

                                date_created:

                                DateTime.parse(
                                date_created),

                                is_archived:
                                is_deleted:

                                )),
                            );
        })),
        IconButton(
            icon: Icon(Icons.archive),
            onPressed: () {
                //print(id);

                DatabaseProvider.db.toArchive(id);

                Fluttertoast.showToast(
                    msg: "Заметка
перемещена в архив",
                    toastLength:
                    Toast.LENGTH_SHORT,
                    gravity:
                    ToastGravity.BOTTOM,

                    fontSize: 16.0);

```

					ПГУ1.090301.24.001 ПЗ	Лист
						120
Изм.	Лист	№ докум.	Подпись	Дата		

```

        setState(() {});
    },
),
IconButton(
    icon: Icon(Icons.delete),
    onPressed: () {
        /*print(id);

DatabaseProvider.db.deleteNote(id);

        setState(() {});*/

DatabaseProvider.db.toDelete(id);

Fluttertoast.showToast(
    msg: "Заметка
перемещена в корзину",
    toastLength:
Toast.LENGTH_SHORT,
    gravity:
ToastGravity.BOTTOM,

    fontSize: 16.0);
    setState(() {});
},
),
Spacer(),
Text(
    textAlign: TextAlign.left,
    DateFormat("dd.MM.yy
HH:MM").format(
DateTime.parse(date_created))),
    SizedBox(
        width: 25,
    ),
],
),
],

```

					ПГУ1.090301.24.001 ПЗ	Лист
						121
Изм.	Лист	№ докум.	Подпись	Дата		

```

        );
    },
),
);
}
}

return SizedBox.shrink();
},
),
floatingActionButton: FloatingActionButton(
    onPressed: () {
        Navigator.pushNamed(context, "/AddNote");
    },
    child: Icon(Icons.note_add),
    backgroundColor: const Color(0xffffcc0dd),
));
}

void _settingModalBottomSheet(context) {
    showModalBottomSheet(
        context: context,
        builder: (BuildContext bc) {
            return StatefulBuilder(
                builder: (BuildContext context, StateSetter
setState) {
                    return SizedBox(
                        child: Column(children: [
                            Padding(
                                padding: EdgeInsets.all(16.0),
                                child: Text(
                                    "Выберите фильтр",
                                    style: const TextStyle(

```

```

        //color: const Color(0xfffc4ca1),
        fontSize: 20.0,
        fontWeight: FontWeight.bold,
    ),
),
),
Column(
  children: <Widget>[
    ListTile(
      title: Text(
        'Работа',
        style: const TextStyle(fontSize:
16.0),
      ),
      onTap: () {
        tag = 'Работа';
        Navigator.of(context).pop();
      },
    ),
    ListTile(
      title: Text(
        'Учеба',
        style: const TextStyle(fontSize:
16.0),
      ),
      onTap: () {
        tag = 'Учеба';
        Navigator.of(context).pop();
      },
    ),
    ListTile(
      title: Text(
        'Личное',
        style: const TextStyle(fontSize:
16.0),

```

					ПГУ1.090301.24.001 ПЗ	Лист
						123
Изм.	Лист	№ докум.	Подпись	Дата		

),
onTap: () {

```

        tag = 'Личное';
        Navigator.of(context).pop();
    },
),
ListTile(
    title: Text(
        'Задачи',
        style:      const      TextStyle(fontSize:
16.0),
    ),
    onTap: () {
        tag = 'Задачи';
        Navigator.of(context).pop();
    },
),
ListTile(
    title: Text(
        'Финансы',
        style:      const      TextStyle(fontSize:
16.0),
    ),
    onTap: () {
        tag = 'Финансы';
        Navigator.of(context).pop();
    },
),
ListTile(
    title: Text(
        'Сбросить фильтр',
        style:      const      TextStyle(fontSize:
16.0),
    ),
    onTap: () {
        tag = null;
        Navigator.of(context).pop();

```

					ПГУ1.090301.24.001 ПЗ	Лист
						125
Изм.	Лист	№ докум.	Подпись	Дата		

} ,

					ПГУ1.090301.24.001 ПЗ	Лист
						126
Изм.	Лист	№ докум.	Подпись	Дата		

```

        ),
      ],
    ),
  ]),
);
},
);
}).then((value) {
  setState(() {});
});
}
}

```

Файл `display_note.dart`

```

import 'package:flutter/material.dart';
import '../model/note_model.dart';

class DisplayNote extends StatelessWidget {
  const DisplayNote({super.key});

  @override
  Widget build(BuildContext context) {
    final NoteModel note =
      ModalRoute.of(context)?.settings.arguments as NoteModel;
    return Scaffold(
      appBar: AppBar(
        title: Text("Заметка"),
        backgroundColor: const Color(0xffffcc0dd),
      ),
      body: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Column(
          children: [
            Text(

```

					ПГУ1.090301.24.001 ПЗ	Лист
						127
Изм.	Лист	№ докум.	Подпись	Дата		

```

        note.title.toString(),
        style: TextStyle(fontSize: 28.8, fontWeight:
FontWeight.bold),
    ),
    SizedBox(
        height: 16.0,
    ),
    Text(
        note.body.toString(),
        style: TextStyle(fontSize: 20.0),
    ),
],
),
),
);
}
}

```

Файл main.dart

```

import 'package:flutter/material.dart';
import 'package:new_notes/screens/archive_notes.dart';
import 'package:new_notes/screens/deleted_notes.dart';
import 'package:new_notes/screens/display_note.dart';
import 'package:new_notes/screens/update_note.dart';
import '../screens/all_notes.dart';
import '../screens/add_note.dart';
import 'model/bottom_bar.dart';

void main() {
    runApp(const MyApp());
}

class MyApp extends StatelessWidget {
    const MyApp({super.key});

```

```

@override
Widget build(BuildContext context) {
  return MaterialApp(
    debugShowCheckedModeBanner: false,
    initialRoute: "/",
    routes: {
      "/AllNotes": (context) => AllNotes(),
      "/AddNote": (context) => AddNote(),
      "/ArchivedNotes": (context) => ArchivedNotes(),
      "/DisplayNote": (context) => DisplayNote(),
      "/DeletedNotes": (context) => DeletedNotes(),
      "/UpdateNote": (context) => UpdateNote(
        id: 0,
        date_created: DateTime.now(),
      ),
    },
    theme: ThemeData(
      colorScheme: ColorScheme.fromSeed(seedColor:
Colors.deepPurple),
      useMaterial3: true,
    ),
    home: const BottomBar(),
  );
}
}

```

Приложение В

(Обязательное)

Стартап-проект. Кроссплатформенное мобильное приложение «Заметки»
Презентация

Выпускная квалификационная работа на тему: «Стартап-проект. Кроссплатформенное мобильное приложение «Заметки»

Автор работы студент группы 20ВВП1 Тельнова А. Д.
Научный руководитель к.т.н., доцент кафедры ВТ Синев М.П.

Пенза, 2024

Актуальность проекта

В настоящее время люди сталкиваются с большим потоком информации и высокой загруженностью. В перенасыщенном информацией мире тяжело держать в голове большие объемы важных сведений, список дел и проблем. При этом значительно снижается концентрация внимания и производительность.

Приложения с возможностью фиксирования важной информации являются эффективным инструментом для организации распорядка дня, четкого распределения времени. Они позволяют зафиксировать все важные аспекты рабочей или личной деятельности, повысить работоспособность и расставлять приоритеты. Электронные записи можно вести в любое время и в любом месте, так как смартфоны стали неотъемлемой частью жизни человека.

Цели и объекты работы

Цель работы – разработать кроссплатформенное мобильное приложение, позволяющее пользователям фиксировать важную информацию и управлять ей.

Объекты работы

Приложение будет предоставлять следующие функции:

- создание заметки;
- редактирование заметки;
- удаление заметки;
- просмотр заметки;
- назначение тегов заметке;
- сортировка заметок;
- управление заметками: перемещение заметки в корзину, восстановление заметки из корзины, перемещение заметки в архив, перемещение заметки в раздел с актуальными записями.

Аналоги

Microsoft OneNote



Достоинства приложения:

- возможность синхронизации мобильного приложения с десктопной и веб-версией приложения;
- возможность создавать записные книжки и разделы в записной книжке;
- возможность прикреплять к записям рисунки от руки.

Недостатки:

- не самый простой функционал;
- нагруженный интерфейс;
- урезанный функционал в сравнении с десктопной версией;
- большой вес приложения.

Изм.	Лист	№ докум.	Подпись	Дата

Аналоги



Достоинства приложения:

- наличие сканера документов;
- возможность подключения Google Календаря;
- синхронизация на разных устройствах;
- создание блокнотов для разграничения информации, создание меток.

Недостатки приложения:

- ограниченность бесплатной версии;
- перегруженность интерфейса;
- ограниченное количество памяти для заметок;
- ограниченное количество устройств для синхронизации;
- большой вес приложения.

Аналоги

Notex



Достоинства приложения:

- простой и удобный интерфейс;
- возможность использовать ключи для шифрования заметок;
- перемещение между заметками свайпом влево/вправо;
- изменение цвета фона заметки.

Недостатки приложения:

- безвозвратное удаление заметок;
- неудобный выход из процесса форматирования текста;
- ограниченность функционала в бесплатной версии;
- упрощенный дизайн

Стек технологий



Android
Studio



Flutter

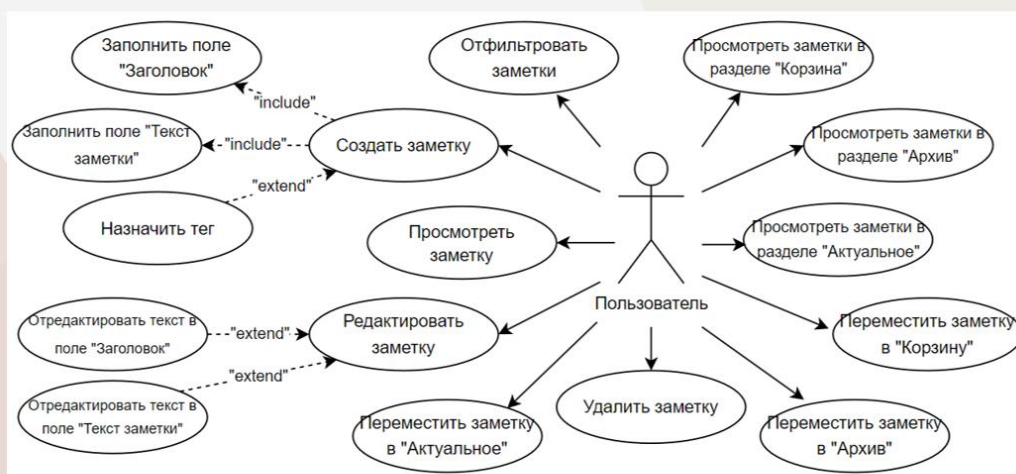


SQLite



Dart

UML-диаграмма вариантов использования



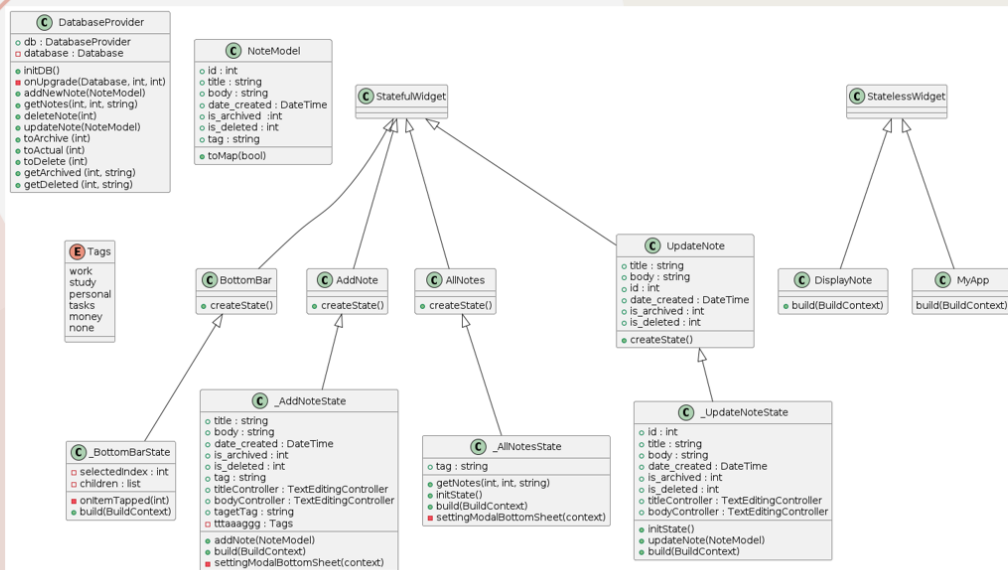
Изм.	Лист	№ докум.	Подпись	Дата

ПГУ1.090301.24.001 ПЗ

Лист

134

UML-диаграмма классов



Карта экранов



Изм.	Лист	№ докум.	Подпись	Дата

ПГУ1.090301.24.001 ПЗ

Лист

135

Сегменты ЦА

Целевая аудитория приложения «Заметки» довольно широкая, так как заметками пользуются люди разных возрастов, профессий и интересов.

Основные группы пользователей приложения «Заметки»:

Студенты и школьники	Бизнесмены и офисные работники	Фрилансеры и удалённые сотрудники	Путешественники и туристы.	Люди, ведущие здоровый образ жизни
Они используют приложение для создания заметок по учёбе, конспектов, планов на будущее и других учебных материалов.	Для них приложение является незаменимым инструментом для ведения дел, планирования встреч, составления отчётов и других рабочих документов.	Эти люди часто работают дома или в общественных местах, поэтому им необходимо иметь под рукой удобный инструмент для создания заметок и напоминаний о важных делах.	Приложение помогает им сохранять информацию о бронировании билетов, гостиниц, экскурсий и других туристических услуг.	В приложении они могут создавать заметки о своём питании, тренировках и других аспектах здорового образа жизни.

Ключевые ресурсы для запуска проекта

- разработчики программного обеспечения;
- SMM-специалисты, занимающиеся продвижением продукта через социальные медиа платформы;
- компьютеры, серверы, программное обеспечение и другое техническое оборудование для разработки и обслуживания системы;
- ресурсы для финансирования разработки, маркетинга, продаж и операционных расходов бизнеса.

Заключение

В результате выполнения выпускной квалификационной работы было разработано кроссплатформенное мобильное приложение «Заметки», обладающее удобным и интуитивно понятным пользовательским интерфейсом.

Данный проект представляет собой приложение, позволяющее записывать важную информацию, список дел, мысли и итоги деловых или личных встреч. Приложение позволяет просматривать зафиксированную информацию, а также эффективно управлять ей. В результате разработки приложение «Заметки» является кроссплатформенным: функционирует на смартфонах на платформах iOS и Android.

Спасибо за внимание!